AD-A068 612   LOGICON INC SAN DIEGO CA TACTICAL AND TRAINING SYSTE--ETC F/G 17/2
SPEECH UNDERSTANDING IN AIR INTERCEPT CONTROLLER TRAINING SYSTE--ETC(U)
JAN 79 M W GRADY, J E PORTER, W J SATZER   N61339-78-C-0044

UNCLASSIFIED   NAVTRAEQUIPC-78-C-0044-1   NL

| OF |

AD
A068612

END
DATE
FILMED
6-79
DDC

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER 18 19 NAVTRAEQUIPCEN 78-C-0044-1	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 SPEECH UNDERSTANDING IN AIR INTERCEPT CONTROLLER TRAINING SYSTEM DESIGN.		5. TYPE OF REPORT & PERIOD COVERED 9 Final Report, ~~Feb. 1978~~ Feb - Nov 1978
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 10 M.W. Grady; J.E. Porter; W.J. Satzer, Jr.; B.D. Sprouse		8. CONTRACT OR GRANT NUMBER(s) 15 N61339-78-C-0044
9. PERFORMING ORGANIZATION NAME AND ADDRESS Logicon, Inc. P.O. Box 80158 San Diego, CA 92138		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 3753-6P1
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Training Equipment Center Code N71 Orlando, FL 32813		12. REPORT DATE 11 January 1979
		13. NUMBER OF PAGES 68
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) 12 73p.		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Speech Recognition
Air Intercept Control
Automated Adaptive Training
Controller Training System
Speech Understanding
Connected Speech Recognition

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the requirements for a speech recognition and understanding system to support an automated training system for Air Intercept Controllers (AIC). A combined isolated word recognition (IWR) and limited connected speech recognition (LCSR) system was developed and tested in a laboratory AIC training system model. Speech stylization constraints were minimized, resulting in particularly challenging recognition requirements. Integration of the IWR and LCSR techniques proved difficult.

UNCLASSIFIED 411 171 B

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Foreword

This is the third in a series of reports leading to the development of a limited continuous speech recognition (LCSR) capability for isolated word recognition (IWR) hardware. This is the first application of the algorithms to a specific, real-world vocabulary. A combination of IWR and LCSR was attempted on the same input phrase in some instances. The resulting system was low in user acceptance. Increased sophistication will be required for further application.

Thanks are extended to the command and staff of the Fleet Combat Training Center, Pacific, San Diego. LCDR Cleveland, OSCS Billups, OSC Lindsay, and Mr. Spencer proved invaluable in the testing of the techniques developed herein.

R Breaux
R. BREAUX
Scientific Officer

TABLE OF CONTENTS

Section		Page
I	INTRODUCTION	5
	Scope	5
	Related Documents	5
	Document Organization	6
II	BACKGROUND AND TASK DEFINITION	7
	Background	7
	Objectives	7
	Nature of the Risks	7
	Approach	8
III	THE AIC VOCABULARY	9
	The Scenarios	9
	Speech Conventions	10
	The Vocabulary Lists	11
IV	VOICE DATA COLLECTION PROGRAMS	15
	IWR Supporting Programs	15
	LCSR Supporting Programs	17
V	SPEECH RECOGNITION PROGRAMS	19
	The VIP Driver	19
	IWR Logic	19
	LCSR Logic	21
	Voice Validation Program	21
VI	RESULTS, CONCLUSIONS, AND RECOMMENDATIONS	22
	Results and Conclusions	22
	Recommendations	27
Appendix A	MINT for AIC	29
Appendix B	Curve Fitting Routine	67

LIST OF TABLES

Table **Page**

1 AIC Vocabulary — Scenarios 1 and 2 (27 Phrases). 12

2 AIC Vocabulary — Scenario 3 (70 Phrases) 13

3 AIC Recognition Lexicon (127 Elements) 14

4 Ten Numbers Used During Voice Training 16

5 LCSR Routines Used for the Generation of a MIND File . . 18

LIST OF ILLUSTRATIONS

Figure **Page**

1 IWR/LCSR Software Organization 20

2 Timing Figures for Two Speakers: G and N 24

3 Timing Figures for Two Speakers: B and L 25

SECTION I

INTRODUCTION

SCOPE

This document is the final technical report for the work performed under contract N61339-78-C-0044: Limited Continuous Speech Recognition (LCSR) for Air Intercept Controller (AIC) Training. The purpose of the study was to identify operational training requirements and limitations of AIC training, identify AIC vocabulary requirements for a dynamic interactive training environment, and develop experimental subsystems for low-cost implementation of an LCSR capability for AIC training. The document is intended for use by the Naval Training Equipment Center's Human Factors Laboratory and other interested parties in support of the definition, specification, and design of an experimental prototype training system for the Air Intercept Controller.

RELATED DOCUMENTS

The following documents describe work which is related to the efforts discussed herein:

a. Use of Computer Speech Understanding in Training: A Demonstration Training System for the Ground Controlled Approach Controller; Technical Report NAVTRAEQUIPCEN 74-C-0048-1; Logicon, Inc., July, 1976.

b. Use of Computer Speech Understanding in Training: A Preliminary Investigation of a Limited Continuous Speech Recognition Capability; Technical Report NAVTRAEQUIPCEN 74-C-0048-2; Logicon, Inc., June, 1977.

c. LISTEN: A System for Recognizing Connected Speech Over Small, Fixed Vocabularies, In Real Time; Technical Report NAVTRAEQUIPCEN 77-C-0096-1; Logicon, Inc., April, 1978.

d. Air Intercept Controller Training: A Preliminary Review; Technical Report NAVTRAEQUIPCEN 77-M-1058-1; Logicon, Inc., June, 1977.

e. NAVTRAEQUIPCEN Specification N-71-277: Specification for Study, Limited Continuous Speech Recognition for Air Intercept Controller Training.

f. A Laboratory Investigation of Requirements for Air Intercept Controller Training; Technical Report NAVTRAEQUIPCEN 78-C-0053-1; Logicon, Inc.; in press.

DOCUMENT ORGANIZATION

Following these introductory remarks, the report discusses in Section II the background against which this study was conducted, and formulates more precisely the problems addressed in the study. Section III describes the AIC vocabulary addressed by the effort. Section IV describes the programs developed for establishing voice reference data, and Section V describes the actual recognition software. The document concludes (Section VI) with a discussion of the results, conclusions and recommendations derived from this study.

SECTION II

BACKGROUND AND TASK DEFINITION

BACKGROUND

During calendar year 1977, the Naval Training Equipment Center sponsored a research and development effort directed toward establishing a real-time, moderate cost system which would recognize connected digits spoken by a speaker for whom the algorithm and data base had been specifically tailored. That R&D activity has just recently been continued. Nevertheless, the initial successes of the adopted approach, together with the need for this capability in an automated training system for Air Intercept Controllers, prompted the government to initiate an (admittedly high risk) study aimed at developing a speech understanding subsystem (SUS) utilizing the new recognition techniques. This SUS would then be tested in a laboratory AIC training model concurrently being developed.

OBJECTIVES

The objectives of this activity have been threefold:

a. Achieve a greater understanding of the demands placed on computer-based speech recognition by an automated AIC training system.

b. Determine if the previously demonstrated connected digit recognition system could be effectively combined with the more usual isolated word recognition (IWR) systems to satisfy the AIC requirements.

c. Provide an applications environment for the continued development of the recognition algorithms to provide focus and ensure the earliest possible realization of an operationally useful recognition capability.

NATURE OF THE RISKS

The objectives for this study outlined in the previous subsection represent problem areas relative to the constraints imposed by near-term speech technology and training contexts for the following reasons:

a. An automated AIC training system featuring objective performance measurement, concept sequencing, and adaptive syllabus control does not exist - in any form - today. While a speech-based automated Ground Controlled Approach (GCA) controller taining system has been investigated, the automated AIC training problem represents a significant advance in both the application of the speech technologies as well as training systems design.

b. Automatic recognition of AIC trainee speech presents a formidable challenge. Unlike the GCA controller's phraseology, the AIC speaks in relatively free flowing format with significant information often embedded within the transmissions. However, the unnatural speech stylization required to make IWR work with the AIC vocabulary is disturbing to the trainee and degrades training effectiveness. Therefore IWR, by itself, would not be an acceptable approach toward satisfying the automatic speech recognition requirements of an AIC training system. A complete LCSR capability is beyond the near-term forecast for the computer based recognition technology. Consequently, a mixed IWR and LCSR system is the only reasonable approach; but it, too, must be examined and described.

c. The demonstrated digit LCSR capability was coded in a stand alone environment in which the computational resources of the minicomputer were not shared with any competing functional tasks. Moreover, the demonstration was entirely context free, simply outputting recognized digits to a CRT for visual display. The design and implementation of an IWR/LCSR subsystem for incorporation into an AIC training system remains a challenging task.

d. The problems associated with automating the collection of voice reference data for the LCSR algorithms are significant. The LCSR capability demonstrated earlier was strictly speaker dependent; that is, long and arduous data collection and processing was required for each speaker. This method is, of course, far from operationally acceptable.

APPROACH

The technical approach taken to satisfy the objectives of this study was:

a. Define the functional AIC vocabulary associated with the learning objectives being addressed in the AIC laboratory model.

b. Establish reasonable stylization constraints and thereafter delineate the recognition lexicon to be addressed by the study.

c. Design a speech understanding system which could recognize the chosen AIC vocabulary with the minimum stylization constraints consistent with the developed technologies.

d. Implement the computer programs required to establish voice reference data and recognize the selected vocabulary.

e. Test the speech understanding subsystem in a laboratory AIC training-like environment.

f. Make recommendations for subsequent recognition capabilities in the experimental prototype AIC training system.

SECTION III

THE AIC VOCABULARY

THE SCENARIOS

Study of the AIC vocabulary centered around three scenarios which were carefully chosen to represent a sufficiently generalized set of AIC tasks. The three scenarios are briefly described in the paragraphs which follow in order to familiarize the reader with the functional environment supported by the recognition capabilities described later.

BASIC INTERCEPTS. Scenario 1 addressed the basic tasks which the AIC must perform in conducting the simplest intercept. As the exercise unfolds, the AIC must locate his assigned aircraft and establish radio contact with the pilot. When a bogey (hostile aircraft) is detected, the AIC communicates with the pilot to vector him to a nearest collision intercept. As the exercise proceeds, the controller must regularly and accurately provide information to the pilot concerning the bogey's bearing, range, heading, and speed. Scenario 1 concludes when the friendly, controlled aircraft comes into radar contact with the hostile aircraft.

REALISTIC INTERCEPTS. Scenario 2 builds upon the basic structure of Scenario 1, adding several complications that more nearly represent actual air intercepts. In addition to providing bogey position and velocity updates, the controller must now also detect and report any sudden changes in the bogey's heading, and recommend new vectors to accommodate the maneuvering hostile aircraft. Moreover, the AIC must detect and report the position and velocity of other aircraft in the vicinity of the controlled aircraft. Finally, the AIC must respond to communications from the pilot at the point of radar contact, at the time when the pilot takes over the intercept himself ("judy"), and when the pilot loses contact with the bogey and needs additional position, velocity, and vectoring information.

THE TRAINING ENVIRONMENT. In addition to learning to control aircraft in combat-like intercept conditions, the operational controller is often called upon to assist in pilot training by setting up mock intercepts in well established training areas. Scenario 3 addresses this training environment and also provides a more challenging vocabulary requirement for the AIC training system. This is particularly true because the transmissions are largely commands rather than advisories. Accurate recognition is therefore essential to the realistic presentation of the scenario. Moreover, there are many more phrases defined for this scenario. Scenario 3 commences with two aircraft flying in formation toward the training area. The AIC makes radar contact with the two aircraft and establishes a lost communications protocol with the pilots. The AIC vectors the aircraft to the area and then maintains the aircraft in the area by providing heading changes. The AIC then detaches one aircraft, who will play the bogey, and turns the other aircraft for separation. The controller determines a planning bearing, target aspect angle, and track crossing angle based upon the point at which he desires the intercept to take place. After getting the proper separation, the AIC turns the aircraft for the mock intercept and Scenario 3 continues as described in Scenario 2. When the aircraft merge, the AIC provides breakaway headings and Scenario 3 concludes as the two aircraft separate.

SPEECH CONVENTIONS

Based upon the three scenarios described in the preceding subsection, the relevant AIC transmissions were identified. This vocabulary, in turn, was studied in combination with the limitations imposed by the technologies and the study objectives of this project. An important area for investigation was the extent to which vocabulary could be dictated to the AIC students. Complete flexibility is neither feasible (from a training system design point of view) nor desirable (from the operational point of view). A training system that demands reasonable standardization is commendable but any unnatural speech constraints must be traded off against user acceptance.

In this regard, certain conventions were adopted for the purposes of this study. Feedback from the AIC training community regarding the acceptability of these conventions is addressed in Section VI. The conventions fall into two categories: general rules and stylization requirements.

GENERAL RULES. Three general rules are reflected in the AIC vocabulary presented herein.

a. The call sign (Snake or Viper) shall be used in conjunction with a transmission if and only if the AIC is transmitting information which requires some action on the part of the pilot. Advisories or responses shall not allow use of the call sign.

b. "Over" may be said at any time, but will always require a short pause (about one second in duration) preceding it.

c. The AIC must never pause preceding a three digit heading, and must always pause following a three digit heading.

STYLIZATION CONSTRAINTS. Because the speech understanding subsystem utilizes isolated word (or phrase) recognition techniques, the user must constrain his verbalizations according to the phrase elements predefined by the system designers. These phrase elements are indicated by semicolons in the tables which follow, and require a short pause during voicing. These stylization requirements (and, for that matter, the general rules as well) could be significantly relaxed in an environment that utilized only the LCSR technology rather than a mixed IWR/LCSR approach.

THE VOCABULARY LISTS

Table 1 presents complete phrases used in Scenarios 1 and 2. XXX is a three digit number between 001 and 360 (bearing or heading), YY is a number between zero and sixty-three (range), and Z is a number between 1 and 9 (speed in tenths of a mach).

Table 2 presents complete phrases used in Scenario 3.

Table 3 presents the recognition lexicon derived from the phrases and stylization constraints defined in Tables 1 and 2.

TABLE 1. AIC VOCABULARY — SCENARIOS 1 AND 2 (27 PHRASES)

Snake radio check
Snake vector XXX; for bogey
Snake port XXX; for bogey
Snake starboard XXX; for bogey
Bogey XXX; YY
Bogey tracking XXX; speed point Z
Stranger XXX; YY
Stranger tracking XXX; speed point Z
Stranger XXX; YY; tracking D

```
D:  North   Northeast
    South   Northwest
    East    Southeast
    West    Southwest
```

Stranger opening
Bogey jinking left
Bogey jinking right
Roger, that is your bogey tracking XXX
Negative your bogey XXX; YY
Say again
Correction
Disregard this transmission
Over
Out
Roger

TABLE 2. AIC VOCABULARY — SCENARIO 3 (70 PHRASES)

Snake radio check

Viper radio check

Snake, mark your TACAN

Roger, radar contact

Snake, say lost communications intentions

Snake, Tango 1, Tango 2 hot;
recommend rendezvous Point Sierra

Viper detach port XXX

Viper detach starboard XXX

$$\text{Viper} \left\{ \begin{matrix} \text{Vector} \\ \text{Port} \\ \text{Starboard} \\ \text{Port Hard} \\ \text{Starboard Hard} \end{matrix} \right\} \text{XXX; as bogey}$$

$$\text{Snake} \left\{ \begin{matrix} \text{Vector} \\ \text{Port} \\ \text{Starboard} \\ \text{Port Hard} \\ \text{Starboard Hard} \end{matrix} \right\} \text{XXX; for bogey}$$

$$\left. \begin{matrix} \text{Snake} \\ \text{Viper} \end{matrix} \right\} \left\{ \begin{matrix} \text{Vector} \\ \text{Port} \\ \text{Starboard} \\ \text{Port Hard} \\ \text{Starboard Hard} \end{matrix} \right\} \text{XXX;} \left\{ \begin{matrix} \text{for the area} \\ \text{for separation} \\ \text{for breakaway} \end{matrix} \right.$$

$$\left. \begin{matrix} \text{Snake} \\ \text{Viper} \end{matrix} \right\} \left\{ \begin{matrix} \text{Continue} \\ \text{Breakaway} \end{matrix} \right\} \text{XXX}$$

$$\left. \begin{matrix} \text{Snake} \\ \text{Viper} \end{matrix} \right\} \text{anchor} \left\{ \begin{matrix} \text{Port} \\ \text{Starboard} \end{matrix} \right.$$

$$\left. \begin{matrix} \text{Snake} \\ \text{Viper} \end{matrix} \right\} \left\{ \begin{matrix} \text{tighten} \\ \text{ease} \end{matrix} \right\} \text{turn}$$

Bogey XXX; YY

Bogey tracking XXX; speed point Z

Roger that is your bogey tracking XXX

Negative your bogey XXX; YY

Say again

Correction

Disregard this transmission

Over

Out

Roger

TABLE 3. AIC RECOGNITION LEXICON (127 ELEMENTS)

1	SNAKE RADIO CHECK
2	SNAKE VECTOR
3	BOGEY
4	BOGEY TRACKING
5	STRANGER
6	STRANGER TRACKING
7	STRANGER OPENING
8	BOGEY JINKING LEFT
9	BOGEY JINKING RIGHT
10	NEGATIVE YOUR BOGEY
11	ROGER THAT IS YOUR BOGEY TRACKING
12	SNAKE PORT
13	SNAKE STARBOARD
14	TRACKING NORTH
15	TRACKING SOUTH
16	TRACKING EAST
17	TRACKING WEST
18	TRACKING NORTHEAST
19	TRACKING NORTHWEST
20	TRACKING SOUTHEAST
21	TRACKING SOUTHWEST
22	SPEED POINT
23	0
24	1
25	2
26	3
27	4
28	5
29	6
30	7
31	8
32	9
33	10
34	11
35	12
36	13
37	14
38	15
39	16
40	17
41	18
42	19
43	20
44	21
45	22
46	23
47	24
48	25
49	26
50	27
51	28
52	29
53	30
54	31
55	32
56	33
57	34
58	35
59	36
60	37
61	38
62	39
63	40
64	41
65	42
66	43
67	44
68	45
69	46
70	47
71	48
72	49
73	50
74	51
75	52
76	53
77	54
78	55
79	56
80	57
81	58
82	59
83	60
84	61
85	62
86	63
87	SNAKE CONTINUE
88	SNAKE BREAKAWAY
89	SNAKE DETACH PORT
90	SNAKE DETACH STARBOARD
91	SNAKE PORT HARD
92	SNAKE STARBOARD HARD
93	SNAKE TIGHTEN TURN
94	SNAKE EASE TURN
95	SNAKE ANCHOR PORT
96	SNAKE ANCHOR STARBOARD
97	VIPER CONTINUE
98	VIPER BREAKAWAY
99	VIPER DETACH PORT
100	VIPER DETACH STARBOARD
101	VIPER PORT
102	VIPER STARBOARD
103	VIPER VECTOR
104	VIPER PORT HARD
105	VIPER STARBOARD HARD
106	VIPER TIGHTEN TURN
107	VIPER EASE TURN
108	VIPER ANCHOR PORT
109	VIPER ANCHOR STARBOARD
110	FOR THE AREA
111	FOR SEPARATION
112	FOR BREAKAWAY
113	FOR BOGEY
114	AS BOGEY
115	CORRECTION
116	SAY AGAIN
117	ROGER RADAR CONTACT
118	SNAKE SAY LOST COMMUNICATIONS INTENTIONS
119	SNAKE TANGO ONE TANGO TWO HOT
120	RECOMMEND RENDEZVOUS POINT SIERRA
121	SNAKE MARK YOUR TACAN
122	VIPER RADIO CHECK
123	ROGER
124	OVER
125	OUT
126	ROGER OUT
127	DISREGARD THIS TRANSMISSION

SECTION IV

VOICE DATA COLLECTION PROGRAMS

A requirement of essentially all speech recognition techniques is to make available to the system information which directly or indirectly reflects the vocabulary to be recognized and the characteristics of the speaker's voice. This information is used as a referent against which the speech signals are compared and classified during the recognition procedure.

This section discusses the programs which are developed for establishing the reference data for the IWR/LCSR speech understanding subsystem. The unique features of the AIC vocabulary necessitated significant modifications to the voice data collection programs used in earlier IWR-based laboratory studies. In addition, computer programs were developed to perform some of the calculations previously done by hand in support of establishing the LCSR reference data base.

IWR SUPPORTING PROGRAMS

The GCA laboratory system utilized a collection of software, the Voice Data Collection (VDC) programs, to create the reference patterns for a user-defined vocabulary. These programs are described in detail in Section VII and Appendix B of NAVTRAEQUIPCEN 74-C-0048-1, cited as reference (a) in Section I of this report. Modifications to this software were required to support the recognition algorithms developed for the AIC vocabulary. These changes are discussed in the following paragraphs.

VOCABULARY LIST GENERATION. The Vocabulary List Generation (VLG) program was modified to enable the creation of vocabulary lists consisting of up to 192 items. The software was updated to Revision 6.3 of Data General's Real-time Disk Operating System, and was re-written in FORTRAN 5.

The only functional change to the VLG program was in the area of pseudo-syntax. A new syntax word was defined which specified:

a. 3 digits must follow this phrase

b. 1 digit must follow this phrase.

c. This phrase is complete in itself.

d. This is a partial phrase.

e. This is a pseudo-item of the lexicon.

COMMAND FILE GENERATION. The Command File Generation (CFG) program was not significantly modified. It was, however, updated to the latest revision of the operating system.

VOICE TRAINING MODE. The actual voice training program required significant modifications to support the unique features of the AIC vocabulary and was rewritten in FORTRAN 5 at the Real-Time Disk Operating System Revision 6.3 level. A description of the functional procedure utilized in creating the reference data will expedite a description of the supporting software.

The IWR phrases were trained in a fashion similar to that used in the GCA laboratory environment. The Vocabulary List file was used with a Command File to direct the prompting of AIC phrases in a semi-realistic order. The prompting and data collection were not integrated into the AIC training however. The most significant departure from the earlier technique was necessitated by phrases such as "bogey tracking 147". In order to minimize the requirements on unnatural stylizations, no pause was required between the heading digits. This required modifications to the data collection program to form an IWR-type reference pattern for the phrases which were followed by three digits.

A set of 10 three-digit numbers were defined which ranged from the shortest three-digit number to the longest, in the range 001 to 360. These numbers are shown in Table 4. The numbers were chosen based upon statistics gathered in the earlier LCSR development work, and assumed the user would say "niner" rather than "nine". These ten numbers were stored as pseudo vocabulary items (items 128 through 137) in the Vocabulary List file, with a syntax word indicating they were not real items of the lexicon. Before going through the voice training procedures, each user was prompted to say these ten numbers, three times each. The length of time (actually, the number of VIP samples) required to voice these numbers was saved, and the average time calculated. This number was input along with the user's name when signing-on to the voice training program.

TABLE 4. TEN NUMBERS USED DURING VOICE TRAINING

118	142	194	255	030
211	173	017	349	096

Each phrase that was followed by a three-digit heading or bearing was joined with each of these ten pseudo-items in the Command File. For example, in training the partial phrases "snake vector" and "for bogey", the Command File entry would prompt "snake vector 118 for bogey". The user was instructed to pause only after the three-digit number; in other words, between "8" and "for" in the example. The raw input data for the first part of the utterance was then truncated by the average three-digit time calculated earlier and used to form a feature pattern representing the partial phrase "snake vector". The raw data from the second part of the utterance (following the pause) was used intact to form a feature pattern representing the partial phrase "for bogey". After ten such patterns were collected for each phrase or partial phrase, a voice reference pattern was formed for each and saved on the disk file.

In addition to the changes necessitated by these heading and bearing phrases, the voice training software was modified to create a slightly different Voice Data (VD) file. Most notably, the reference patterns were not arranged according to their time length as was the case in the earlier GCA system. Instead, the patterns were stored by vocabulary index number. The VD file was also modified to represent up to 192 vocabulary items.

LCSR SUPPORTING PROGRAMS

No significant changes were made to the programs previously developed to create the LCSR reference data. (The sequence of programs that were executed to form the LCSR reference data file is shown in Table 5.) One additional program was developed. In the earlier LCSR efforts, certain parameters were determined by manually fitting curves to the observed data. This "curve fitting" procedure was automated by the new program, thus decreasing the amount of hand analysis required in the reference data generation process. (See Appendix B.)

TABLE 5. LCSR ROUTINES USED FOR THE GENERATION OF A MIND FILE

1. EXTRACT — digitizes, compresses, and stores voice inputs
2. GWIZ — lists utterances and makes a first cut at marking the words within the utterance
3. MEND — creates the example spaces from the handcut input data
4. GZEC — forms the sets of transition letter sets
5. RESCUE — retrieves the desired transition letter sets
6. LOOPER — forms the sets of loop letter sets
7. DCARDZ — generates card images for machine formatting
8. MACFOR — produces the formatted machines which include the sets of transition and loop letter sets
9. REVEXA — collects counter data statistics from training speech data
10. VERIFY — checks for human errors in construction of RVCARDS file of counter data to be kept
11. RVDIT — creates separate counter data files for each of the machines from the "good" data
12. COVERT — gathers statistics on the counters and computes the covariance matrix
13. INVERT — inverts the covariance matrix
14. CROAK — prints the dispersion matrix and computes additional statistics
15. REVEX — revised research machine exerciser: exercises the machines over interim test data
16. SQUISH — performs curve fitting funtion, fitting a curve through observed cumulative distribution Q_T quality function points
17. QDFIT — fits a parabola to the set of computed log likelihood ratio values by a least squares procedure
18. ADDER — creates violation tables for transition and loop letter set violations
19. AVRAJ — obtains mean word lengths from counter data file
20. CRAP — generates critical association parameters for GAPSTER
21. GAPSTER — determination of association, gap, and delay values, Δt_a, Δt_s, Δt_{sm}
22. DEALER — generates the file MIND.VD

SECTION V

SPEECH RECOGNITION PROGRAMS

Essentially all of the speech recognition logic was developed especially for this effort. The codes exist as a FORTRAN 5 program designed to run in the foreground of a 96K-word Eclipse computer. Within the AIC environment, the recognition computer shares its resources with the Radar Simulation and Naval Tactical Data System simulation logic running in the background. A Megatek display subsystem is driven by this computer as well. An inter-ground communications area was defined, through which recognition decisions were passed to the pilot model and performance measurement subsystem via the Inter-Processor Bus (IPB).

The general information flow within the speech recognition logic is shown in Figure 1. MEX, the word spotting portion of the LCSR logic, runs in parallel to the IWR algorithm. MINT, the word selection portion of the LCSR logic, is invoked if the IWR logic determines that the last recognized phrase has digits associated with it.

The following subsections describe in greater detail the unique aspects of this speech recognition logic which distinguish it from the IWR software used in the GCA laboratory system and the stand-alone LCSR system.

THE VIP DRIVER

The interface between the Threshold Technology 500 Preprocessor and the recognition software is a user-defined device driver, VIPDR. In the AIC environment, this driver must satisfy the requirements of both the IWR logic and the LCSR logic. Consequently, VIPDR was designed and written to fill one of two input buffers with the raw VIP data, and simultaneously create LCSR-type letters and counts. These were sent on to the MEX algorithm. The particular features to be used in creating the LCSR letters were defined by a software mask to facilitate modifications.

IWR LOGIC

When the LP_4 feature was set indicating the end of a phrase, the IWR logic was initiated. Because the unknown phrase could be either a complete phrase (e.g., "snake radio check") or a phrase with digits ("snake vector 123"), two input feature patterns were created. One pattern represented all of the input data; the other was formed by truncating the raw data by the average length of three digits, determined in the data collection procedure. Each reference pattern was then compared with the appropriate input pattern, depending upon the syntax word.

The correlation algorithm was the same used in previous studies. The comparison routine was extensively re-written, however, to realize the potential for high speed calculations in the AIC hardware environment.

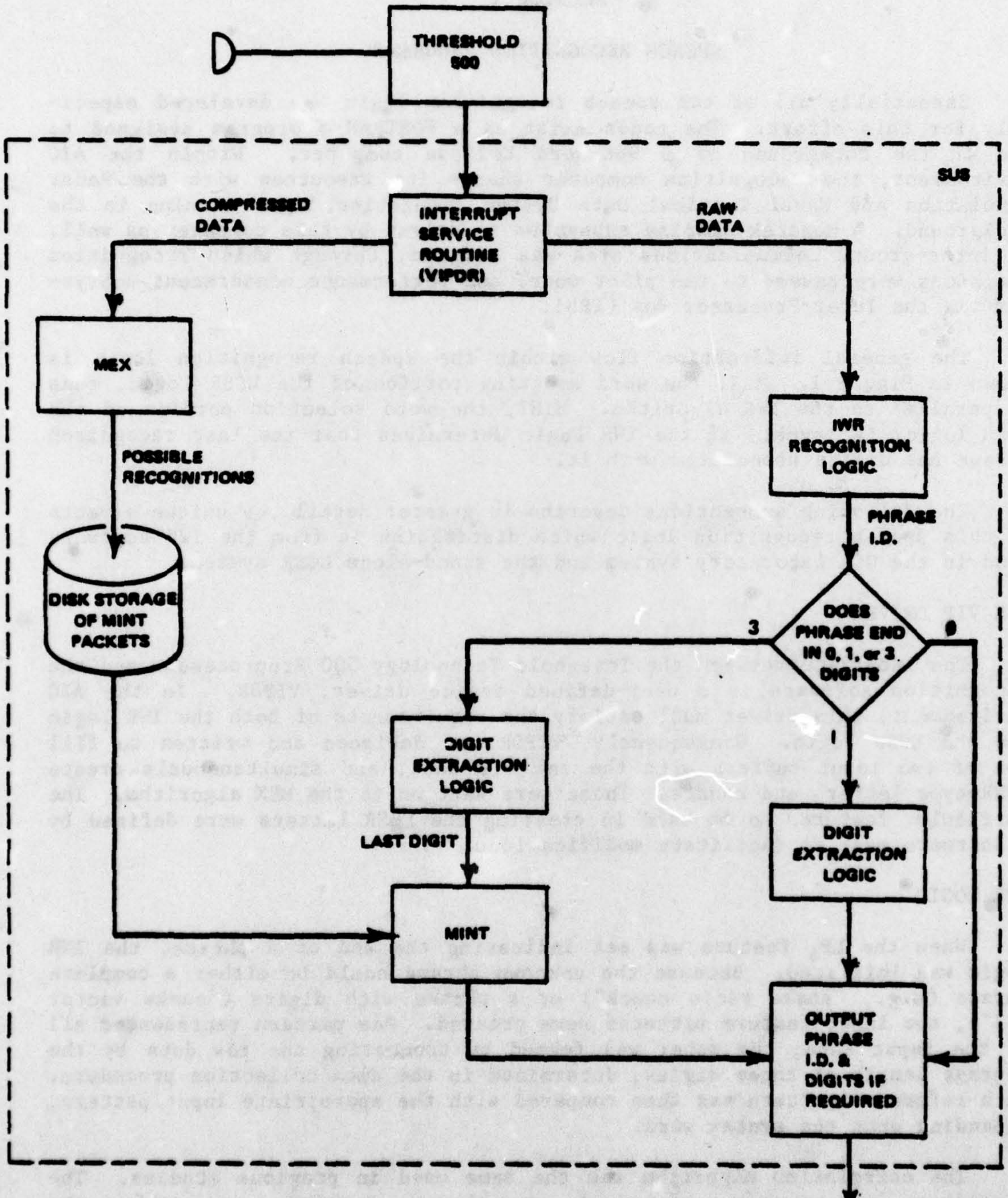


Figure 1. IWR/LCSR Software Organization

Eclipse instructions, the Eclipse hardware stack, and the High Speed Correlator on the VIP interface board were utilized in the new comparison logic.

The most significant change to the IWR software was made in relation to the storage of the reference patterns during recognition. In the earlier GCA studies, the reference patterns were dynamically retrieved from the disc in real-time. In the AIC environment, the reference patterns are stored in extended memory and accessed by the comparison logic via window mapping. Thus, the reference patterns are stored outside the 32K address space accessible by the logic. When the patterns are needed, the desired memory blocks are remapped into the normal address space by enabling the hardware memory management unit. No true data transfer occurs. It is only the addresses at which the data are found that are changed. This is turn enables very high speed data access.

LCSR LOGIC

The AIC environment significantly limits the range of reasonable utterances which are subject to the LCSR procedure. In the laboratory scenarios, only three-digit numbers from 001 through 360 need be recognized. Also, the last digit can be determined using the usually highly accurate "digit extraction" techniques developed for GCA applications. Finally, the numbers are always preceded by non-digit voicings, such as "snake vector".

This supportive information was used in a reformulation of the machine interaction (MINT) algorithm. A detailed description of those changes is presented in Appendix A. The MINT procedure was modified in accordance with the changes and integrated into the AIC speech recognition software. No significant changes were made to the MEX algorithm or software to support the AIC laboratory requirements.

VOICE VALIDATION PROGRAM

The speech recognition logic described above was utilized in the AIC laboratory model. In addition, the software was configured as a stand-alone program and thus served as the framework for a voice data validation program. In this configuraiton, the recognized phrases were simply echoed on the system terminal. As such, this program replaced the validation mode of the GCA Voice Data Collection (VDC) program.

SECTION VI

RESULTS, CONCLUSIONS, AND RECOMMENDATIONS

RESULTS AND CONCLUSIONS

The development of the speech recognition subsystem for the AIC training environment has been a fruitful experience. Various lessons were learned and conclusions were drawn that will influence subsequent implementations. These results and conclusions based upon utilization of the laboratory system are summarized in the following paragraphs.

IMPLEMENTATION EFFORT. Despite the fact that very few new algorithms were designed, the effort needed to implement the AIC speech recognition programs was considerable. The modifications to run in the FORTRAN 5, mapped environment uncovered various unknown features of the operating system and run-time libraries.

MEMORY REQUIREMENTS. The AIC recognition logic shared system resources with the display programs. In order to maintain the high speed processing needed by both recognition and display, it was necessary to add additional memory to the computer. MOS memory was expanded by 32K words and integration efforts were simplified by having the display and recognition software run in separate grounds. The additional cost for the memory was more than compensated for by decreased software development and integration costs. The use of virtual overlays and window mapping enabled the recognition software to effectively use the additional memory.

STYLIZATION CONSTRAINTS. The general rules and stylization constraints imposed upon the AIC vocabulary were not viewed as detrimental to AIC training. This view was shared by all three AIC instructors who were asked to evaluate the chosen vocabulary. It is interesting to note, however, that all three instructors had initial difficulty in putting the constraints into practice. Unlike the GCA instructors, the AIC instructors tended to add pauses rather than delete them. Naive users (those not familiar with the AIC environment) had no difficulty conforming to the chosen vocabulary.

EXPOSURE TO THE SYSTEM. The first exposure that the users of the system had to speech recognition was with the aid of the stand-alone Threshold Technology word recognition system (V19A.SV). The AIC instructors configured the system to recognize just the digits, and were then encouraged to simply play with the system to see what it could and could not recognize. The instructors were purposely left alone during this exposure period so that they would feel comfortable in speaking freely to the machine. This initial period gave them the confidence that their speech really could be recognized by the computer.

Following this digit recognition, ten AIC Phrases were input to the stand-alone program and the users experimented with this small AIC vocabulary. The goal here was to impress upon the instructors that strict conformance to the defined vocabulary was an essential element for good recognition.

IWR TRAINING PROCEDURE. The AIC IWR vocabulary was trained in three separate sessions: (1) range calls (1-63); (2) scenarios 1 and 2 phrases; (3) scenario 3 phrases. The three-digit time data was also collected in session 1 for subsequent use in session 2. Because of complications in achieving good accuracy on those phrases consisting of three-digit numbers (see below), the two intructors from the Fleet Combat Training Center, Pacific, did not proceed into scenario 3 (session 3). (The results and conclusions discussed herein are therefore based on the experience gained in the simpler scenarios 1 and 2. No qualitative difference in recognition requirements exists in scenario 3, but because of the larger vocabulary, it is reasonable to expect poorer results for scenario 3 than achieved in scenarios 1 and 2.)

Data collected from four speakers on the 10 three-digit numbers is shown in Figures 2 and 3. Speaker G is very experienced with speech recognition systems and typically achieves high accuracy. Note the consistency in the three repetitions of each number. Speakers L and B are inexperienced relative to computer speech recognition systems; notice the inconsistency in their data. This data suggests that time may be a valid metric for an automatic algorithm to determine when voice data collections should begin and/or terminate.

Unexpected and severe difficulty was encountered with accurately recognizing phrases which include three digits at the end. Phrases were typically misrecognized with the substitution error not intuitively understandable: "bogey 1 2 3" was misrecognized as "bogey jinking left". Closer analysis suggested that too much raw data was being discarded from the end of the phrase. Several changes were made, culminating in the formation and comparison of five feature arrays. Still, good accuracy was never achieved. The conclusion drawn from this result is that the feature array and reference array are very sensitive to the three-digit heading or bearing voicing, and the usual IWR algorithm is not suitable for determining the beginning portion of the utterance.

J. Porter, who developed the LCSR algorithm, has suggested a dynamic programming algorithm which, he believes, would result in improved accuracy of the initial portion of the utterance. Alternatively, one can consider expanding upon the stylization rules to require a pause both before and after the three-digit numbers.

LCSR TRAINING PROCEDURE. The LCSR vocabulary was trained using the "magic number sets" developed in the earlier studies. This procedure was thus even more divorced from the operational AIC environment than the IWR training procedure. Eighteen number sets were recorded for one speaker (LHN). Twelve sets were used to create the reference data and six sets were saved for subsequent use in the LCSR development effort.

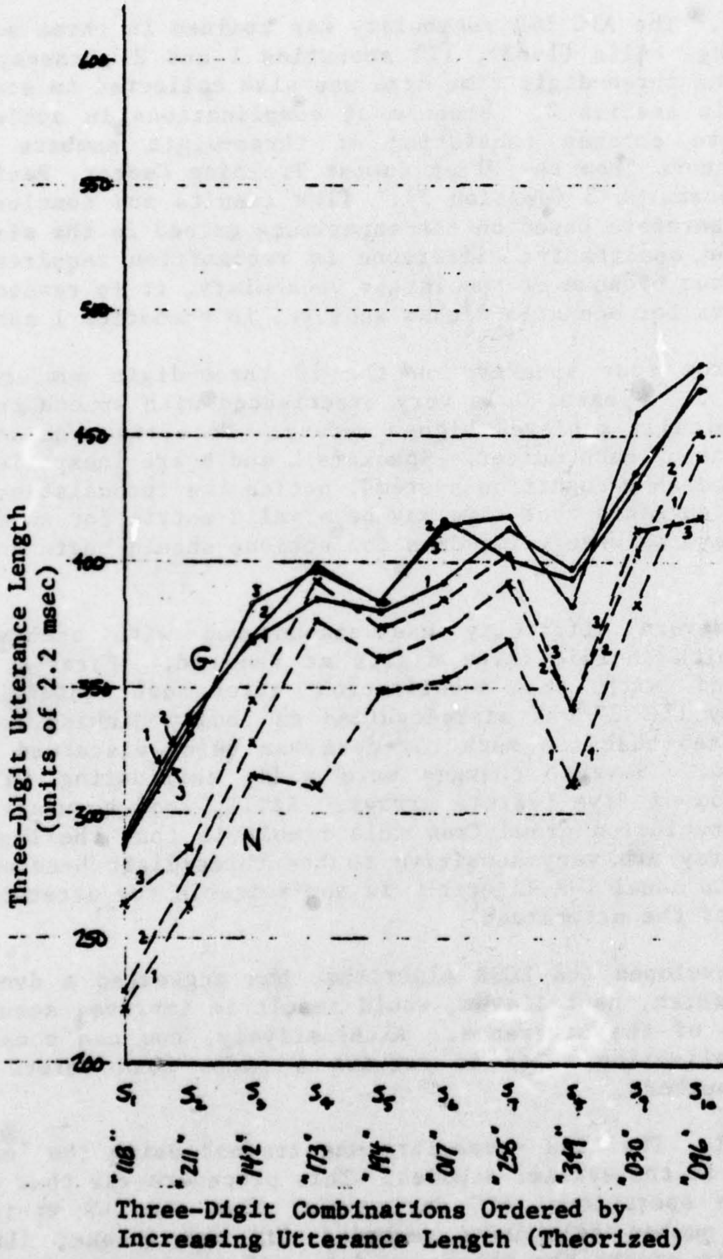


Figure 2. Timing Figures for Two speakers: G and N

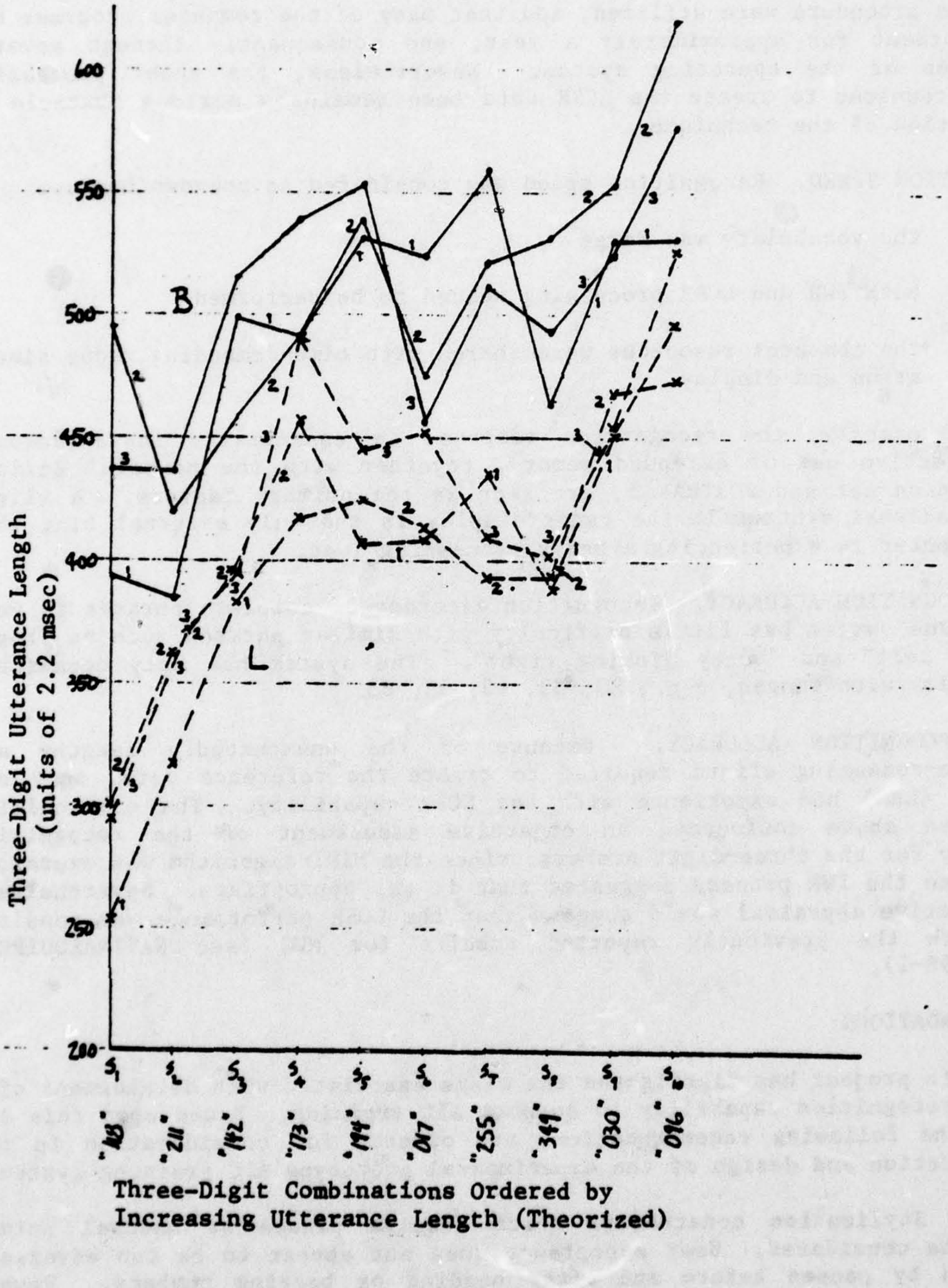


Figure 3. Timing Figures for Two Speakers: B and L

The processing to create just one file for one speaker required two calendar months and three labor months. The procedure was significantly complicated by the facts that personnel who had not previously been involved with the procedure were utilized, and that many of the computer programs had been dormant for approximately a year, and consequently through several revisions of the operating system. Nevertheless, the sheer processing burden required to create the LCSR data base remains a serious obstacle to utilization of the technique.

RECOGNITION SPEED. Recognition speed was considered an unknown because:

a. the vocabulary was large

b. both IWR and LCSR processing needed to be performed

c. the computer resources were shared with time-demanding radar simulation and display

Fortunately, the recognition response is essentially instantaneous. The effective use of extended memory, together with the powerful Eclipse instruction set and FORTRAN 5, are seen as the primary factors. A slight and occasional stutter in the radar display is the only external hint that the computer is experiencing a heavy processing load.

IWR RECOGNITION ACCURACY. Recognition accuracy of complete phrases is very good. The system has little difficulty with similar phrases such as "bogey jinking left" and "bogey jinking right". The system has only occasional difficulty with ranges, e.g., 23, 33, 43, 53, 63.

LCSR RECOGNITION ACCURACY. Because of the unexpectedly lengthy and resource-consuming effort required to create the reference data, only one speaker (LHN) had experience with the LCSR capability. The difficulties discussed above confounded an objective assessment of the recognition accuracy for the three-digit numbers, since the MINT algorithm was exercised only when the IWR process suggested that it was appropriate. Nevertheless, a subjective appraisal would suggest that the LCSR performance was consistent with the previously reported results for MWG (see NAVTRAEQUIPCEN 77-C-0096-1).

RECOMMENDATIONS

This project has highlighted the risks associated with development of a speech recognition capability to support AIC training. Based upon this effort, the following recommendations are offered for consideration in the specification and design of the experimental prototype AIC training system:

a. Stylization constraints which require pauses at natural points should be considered. User acceptance does not appear to be too adversely impacted by pauses before and after heading or bearing numbers. Pauses between each digit are not acceptable, however.

b. Special processing should be developed to recognize those phrases where IWR and LCSR techniques are both required. A dynamic programming based algorithm may be the solution.

c. The reference data generation problem severely restricts the application of the existing LCSR techniques to practical applications. Additional research should be directed toward determining if an algorithm which utilizes only a very small training sample (say ten repetitions) can be developed. Alternatively, some form of speaker adaptation to a previously defined data base may be appropriate. In any event, continued R&D is required if the current LCSR technique is to be useful in the prototype training system.

d. Other connected speech algorithms (such as the earlier Logicon empirical approach, or the Nippon Electric Company DP-100 system) should be considered in designing the AIC prototype.

e. An initial exposure period should be designed into the system to teach users how to speak for optimum recognition accuracy.

f. Speech data collection must be an integral part of the training process.

APPENDIX A
MINT FOR AIC

Review of the MINT Problem

The unpublished memo "Reformulation of the Statistical Basis for the Machine Interaction Problem" contains an accurate mathematical statement of the problem solved by MINT. It is largely applicable to MINT for AIC, so is reviewed here. The reader is also referred to Technical Report NAVTRAEQUIPCEN 77-C-0096-1 for a discussion of the MINT algorithm. The point of departure is to think of the set of potential recognitions picked out of an utterance by MEX as a set of nodes, or abstract points. The time of occurrence of these potential recognitions induces constraints on which potentially recognized words precede which others. The nodes, together with the relation corresponding to the notion "is a potential immediate predecessor," then form a directed graph. It is convenient to append two special nodes called "Start" and "End" to the set of nodes, and to extend the potential predecessor relation to include "is a potential first word" and "is a potential last word" in obvious ways.

The primary role of MEX is thus considered to be recognition of the start and end of the utterance, notification of potential recognitions during the utterance, and recording start times and end times of potential recognitions by which pairs related by the extended relation "is a potential predecessor of" can be identified. These data create the directed graph discussed in the previous report. In addition, MEX provides data to be associated with the nodes and edges of this graph, making it an annotated directed graph.

For each edge of the graph an associated time delay or overlap value can be computed from the recognition start or end times of the nodes at which it is incident. A single numerical value thus annotates each edge. The annotation of each node is more complicated, as it consists of three data:

i) The type of machine responsible for the potential recognition (with dummy values for the Start and End nodes).

ii) An "intrinsic property" value. The exact nature of this value is secondary to the fact that certain conditional probabilities are known about it, but to be concrete we note in passing that LISTEN currently observes a T counter statistic, an L counter statistic and a violation category. The intrinsic property value can therefore now be construed to be an element of the Cartesian product of the sets of all T counter statistics values, L counter statistic values and violation categories.

iii) A set of associated machine types. As described elsewhere, when one potential recognition overlaps another for a (machine-type dependent) sufficient length of time, one potential recognition is said to be "associated" with the other. MEX supplies the data whereby it can be determined, for each potential recognition, what <u>machine types</u> caused associated potential recognitions.

The setting of the MINT problem then consists of a directed graph with annotations, the latter consisting of a numerical value for each edge and three information elements for each node. The memo cited above points out the importance of correctly identifying the setting of the problem (that which is simply "given") and distinguishing the setting from the observation. The following mathematization is useful in what follows, and it makes the important distinction quite clear. There are given:

1) A set of potential recognitions, $\Pi = \{\pi_1, \pi_2, \ldots \pi_n\}$
2) A set of (real) machine types, M
3) A set of nodes $N = \{Start, End\} \cup \Pi$
4) A set of extended machine types, $M' = \{Start, End\} \cup M$
5) A relation E ("potential predecessor" extended to all of N) or N. The members of this relation, $E \subset N \times N$ make the pair G = (N,E) a directed graph. The time, constraints used in defining the potential predecessor relation E, and the utterances we deal with, have properties which guarantee that G is acyclic and connected in the sense that there is a path in G from Start to any other node, and a path from every node (other than End) to End.
6) A function $m: N \to M'$ m(n) is the machine type responsible for the potential recognition n when $n \in \Pi$, else m(n) is Start or End.
7) A set of possible intrinsic properties, I.

An "observation" is then the collection of annotations other than the machine-type identifier. (That is the critical point in the cited memo - it is hopeless to compute the probability of occurrence of individual potential recognitions in a particular order - the structure of the set of possibilities is too complicated. So we take the collection of potential recognitions and their temporal relationships as simply given, then define the hypotheses and observation in terms of these data, thereby avoiding the problem of computing the probability of occurrence of the graph itself.) In mathematical terms:

An observation $\Omega = (\mathcal{D}, \mathcal{I}, \mathcal{A})$ is a set of three functions:

1) $\mathcal{D}: E \rightarrow \mathbb{J}$, the integer-valued delay, overlap or gap associated with each edge.
2) $\mathcal{I}: \Pi \rightarrow I$, the intrinsic properties of each potential recognition (observed by MEX).
3) $\mathcal{A}: \Pi \rightarrow \mathcal{P}(M)$, the set of machine types causing potential recognitions associated with each given potential recognition:

$$\mu \in \mathcal{A}(\pi_i) \Leftrightarrow \exists j \neq i \, [m(\pi_j) = \mu \text{ and } \pi_j \text{ is associated with } \pi_i].$$

Recall π_j is associated with π_i if the overlap in their periods of detection exceeds a criterion c_i, a function of $m(\pi_i)$

The set of hypotheses, $\mathcal{H}$, (hypothetical utterances which were spoken) is the set of all paths in G from Start to End. To each such path there corresponds an unambiguously determined sequence of machine types, and hence of vocabulary items. Note that two or more paths may correspond to the same sequence of hypothetically spoken vocabulary items, so it is not strictly correct to identify hypotheses with individual utterances, as there is really a many-one correspondence between these two sets. Note also that no hypothesis is considered which doesn't have a corresponding path through the graph. The effect of this is that MEX must have a machine of the proper type going to recognition at roughly the correct time in order for MINT to even consider the correct explanation of an utterance.

We assume that there is a single hypothesis which is in fact true; i.e., that there is a unique sequence of potential recognitions forming a path from Start to End, each of which is a real recognition, all others being artifacts. Strict validity of this assumption is questionable, but it can be argued that the single true hypothesis model can be made to describe the facts quite accurately by properly interpreting and computing the statistical parameters needed to solve the problem.

The problem MINT solves is the selection of the hypothesis which best explains the utterance, in the sense that it is the most probable explanation. Bayes' theorem is invoked indicating that, for any $H \in \mathcal{H}$,

$$\text{Prob}(H|\Omega) = \frac{\text{Prob}(H)\,\text{Prob}(\Omega|H)}{\text{Prob}(\Omega)}$$

(In words, this says that the probability that any sequence of potential recognitions is in fact the real sequence, in view of the observation, is the product of the probability that that sequence of recognitions will arise at all, and the probability that the observed details will occur when that sequence is really spoken, divided by the probability that this particular set of observations will ever occur).

In view of the fact that the denominator in the expression above is the same for all hypotheses $H\epsilon\mathcal{H}$, the most probable hypothesis is the one which maximizes the product of a priori probability and conditional probability of occurrence of the observation, i.e., the numerator in the expression above.

A particularly pleasing aspect of this approach to the problem arises when one imposes a frequency-of-occurrence interpretation on the probabilities in the expression above. Under this interpretation each of the probabilities is considered to be a "fraction" of many, many cases, with the delightful result that picking the hypothesis which maximizes the expression on the left is identical, in the long run, with picking the hypothesis which is most often correct.

Really, of course, one cannot compute the product of probabilities in the numerator and thus solve the problem rigorously. One can only estimate those probabilities using models developed from many observations of vocal behavior. And the models must be simple enough to admit the necessary calculation and also the fitting of the model to the voice training data. As an example of the kind of simplifying assumptions which get imposed are the following:

1. The inter-word delays and the association and intrinsic characteristics the potential recognitions are independent, so

$$\mathrm{Prob}(\Omega|H) = \mathrm{Prob}(\mathcal{D}|H)\,\mathrm{Prob}(\mathcal{I}|H)\,\mathrm{Prob}(\mathcal{A}|H)$$

2. The inter-word delays are independent of each other, so

$$\mathrm{Prob}(\mathcal{D}|H) = \prod_{e \in E} \mathrm{Prob}(\mathcal{D}(e)|H)$$

3. The intrinsic characteristics of each potential recognition are independent of each other, as are their associations, so

$$\mathrm{Prob}(\mathcal{I}|H) = \prod_{\pi \in \Pi} \mathrm{Prob}(\mathcal{I}(\pi)|H)$$

and

$$\mathrm{Prob}(\mathcal{A}|H) = \prod_{\pi \in \Pi} \mathrm{Prob}(\mathcal{A}(\pi)|H)$$

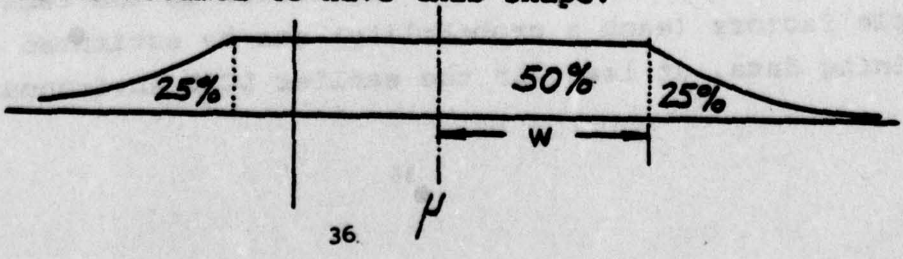
The assumption of statistical independence is imposed several more times, causing each factor in the products above to be further reduced to a product of simpler terms. Finally the needed product of the a priori probability of H and the conditional probability of the observation given the observation is expressed as a product of very many simple terms. Equally important as the simplicity of this expression is the fact that those simple factors (each a probability) can be estimated from the training data, at least in the earlier LCSR environment. Almost all of this previous development will carry over into the AIC environment, so it will be reviewed here before the unique features of the AIC problem are explored.

A. Prob($\mathcal{D}$(e)|H)

Let $e=(n_i,n_j)$ be an edge of the graph G. That is, let n_i be a potential predecessor of the node n_j. Then if n_i and n_j are potential recognitions, $\mathcal{D}$(e) is the delay between the recognition time of n_i and the start of recognition of n_j. If n_i is Start, $\mathcal{D}$(e) is the start delay, and if n_j is end, $\mathcal{D}$(e) is the end delay. (If n_i is Start and n_j is End, $\mathcal{D}$(e) is the length of the utterance.)

We distinguish only two cases; either the hypothesis H contains the edge e or it doesn't; that is, either n_i and n_j are consecutive real recognitions (or n_i is Start and n_j is real, or n_i is real and n_j is End) or they aren't. No distinction is made within the second alternative, so the delay between a real recognition and an artifactual recognition, for example, is not distinguished from the delay between two artifactual recognitions. Beyond that, we do let the distribution of delays depend upon the types of machines responsible for the nodes it joins. Three particular cases are considered.

i) When the nodes n_i and n_j are potential recognitions (neither start nor end) the distribution of delay values is assumed to have this shape:



so $$\mathrm{Prob}(\mathcal{D}(e)=d|H)=\begin{cases}\frac{1}{4w} \text{ if } |d-\mu| \le w\\ \frac{1}{4w}\, e^{-\left(\frac{|d-\mu|}{w}-1\right)} \text{ if } |d-\mu|>w\end{cases}$$

The constants μ and w depend upon the machine types of n_i and n_j, and whether e is an edge in H or not.

ii) When n_i is Start (and $n_j \in \Pi$) the distribution of delay values is assumed to have a mass concentration at zero, and an exponential distribution over values of $d \geq 2$. (The value d=1 is given zero probability of occurrence because the preprocessor eliminates all sounds with a duration of 1 count.) Thus

$$\mathrm{Prob}(\mathcal{D}(e)=d|H)=\begin{cases}p_o \text{ if } d=0\\ \lambda(1-p_o)e^{-\lambda(d-2)} \text{ if } d\geq 2.\end{cases}$$

and the constants p_o and λ depend upon the machine type of n_j and whether or not e is in H.

iii) When n_j is End (and $n_i \in \Pi$) the distribution of delays is assumed to have the same truncated symmetric exponential shape as case i) if the edge is in H, and a uniform distribution over an interval if the edge is not in H. The non-negative character of end delay values requires that a special normalizing factor be added in the first case. Thus:

$$\mathrm{Prob}(\mathcal{D}(e)=d|H)=\begin{cases}\left.\begin{matrix}\frac{m}{4w} \text{ if } |d-\mu| \le w\\ \frac{m}{4w}\, e^{-\left(\frac{|d-\mu|}{w}-1\right)}\end{matrix}\right\} \text{ if } e\in H\\ \left.\begin{matrix}\frac{1}{2w'} \text{ if } |d-\mu'| \le w'\\ 0 \text{ if } |d-\mu'|>w'\end{matrix}\right\} \text{ if } e\notin H\end{cases}$$

where W, W', μ, μ' are functions of the machine type responsible for the potential recognition n_i, and m is a function of w and μ.

B. Prob $(d(\pi)\,|\,H)$

The intrinsic properties of each potential recognition observed or measured by MEX are the violation category, T-counter history and L-counter history. The violation category, V, is a classification of the type of violations which occurred in the process of generating the potential recognition, and contains options such as no violation, violation of a single internal transition letter-set at critical features, violation of a single loop letter set, etc. Denote the set of violation categories as $\mathcal{Q}$.

The T-counter history observed by MEX is summarized in the modified-Mahalanobis metric Q_T, which is a measure of how peculiar the observed T-counter history is as compared to the training data. Q_T is a non-negative real number, and therein lies a problem for this discussion. The distribution of Q_T values is most plausibly modeled as a continuous one with a probability density. For such a distribution, the probability of occurrence of any specific value is zero, so the application of Bayes' rule stated above (in terms of probability of occurrence of the observation, which includes a particular Q_T value) breaks down. One rigorous correction of this state of affairs is to restate Bayes' rule in terms of probability densities for the continuous parts of the problem, but that leads to a fragmented, example-specific characterization. Another is to use Radon-Nikodym derivatives (generalized densities) to recover the uniform treatment, but that is a bit too abstract for this environment.

The formulation we shall adopt here avoids the problem of continuous distributions by assuming that the T-counter metric Q_T takes on values in a countable set $\mathcal{J}$, rather than on $\mathbb{R}^+$. (Since we are dealing with values computed in a computer, this assumption could be argued to be closer to the truth.) The probability of occurrence of particular values from the set $\mathcal{J}$ is assumed to always be non-zero, alleviating the problem with Bayes' theorem. Conceptually the members of $\mathcal{J}$ can be thought of as (usually) small intervals on $\mathbb{R}^+$. We shall not be explicit about the size and location of the intervals, except that they should obviously form a partition of $\mathbb{R}^+$.

A similar, but even worse, problem holds for the L-counter history summary. The summary in this case is a non-negative value, Q_L, most reasonably described as a non-negative real value. Unlike the Q_T values, there is a definite concentration of cases at one particular value of Q_L; viz zero, so Q_L doesn't have a density. Fortunately the same solution works; we assume the Q_L values are taken from a countable set $\mathcal{L}$, such that each element has non-zero probability of occurrence. One element of $\mathcal{L}$ is the single value zero. Other elements of $\mathcal{L}$ can be identified with intervals of $\mathbb{R}^+$. Using the artifice just introduced, the probability that a particular set of intrinsic characteristics $q \in \mathcal{Q}, t \in \mathcal{J}$ and $\ell \in \mathcal{L}$ will be observed for the potential recognition π is assumed to be

$$\mathrm{Prob}(\mathcal{A}(\pi)|H) = \mathrm{Prob}(V(\pi)=q|H)\,\mathrm{Prob}(Q_T(\pi)=t|H)\,\mathrm{Prob}(Q_L(\pi)=\ell|H)$$

$$= I(q)\,J(t)\,K(\ell)$$

reflecting the assumption that the three characteristics (quality category,) Q_T and Q_L are all independent. The functions I, J, K above are probability distributions on the sets $\mathcal{Q}$, $\mathcal{T}$ and $\mathcal{L}$ and are assumed to depend upon whether or not π is in H. The functions J and K also depend upon the type of machine responsible for π.

C. Prob $(\mathcal{A}(\pi)|H)$.

Recall that $\mathcal{A}(\pi)$ is a collection of machine types (a subset of M). Machine type m is a member of $\mathcal{A}(\pi)$ if there is some other potential recognition, π', which overlaps the potential recognition π in time sufficiently to establish π' as associated with π, and m is the machine type of π'. A recognition by a particular type of machine is assumed to have various probabilities of having associated recognitions by various machine types. As usual, we assume these possibilities are independent, and depend only on the machine type m, the machine type responsible for π and whether or not π is real, i.e., $\pi \in H$.
Thus

$$\mathrm{Prob}(\mathcal{A}(\pi)|H) = \prod_{m\in\mathcal{A}(\pi)} A(m) \prod_{m\notin\mathcal{A}(\pi)} (1-A(m))$$

where A is a set of probabilities on M, depending upon the machine type responsible for π and whether or not $\pi \in H$.

D. Prob(H)

This term is the a priori probability that the hypothesis H is true. It answers the following question: Given the directed graph of potential recognitions, what is the probability that the particular path H from Start to End is the set of real recognitions, irrespective of the delay data, instrinsic recognition characteristics and association data? Under the frequency interpretation of probabilities, this number also has the following significance. Suppose MEX is exercised an arbitrarily large number of times. Consider those cases where the timing and the set of potential recognitions is represented by the given directed graph (without the delay, intrinsic characteristic and association data annotated). Then Prob(H) is the fraction of those cases in which the hypothesis H is in fact the correct path through the graph.

Estimation of the a priori probability that H is the correct hypothesis is a difficult issue. The difficulty arises because of the conditioning requirement that the directed graph of the utterance should be the one which in fact was observed. For a given operational environment it should be possible to estimate the probability that a particular sequence of vocabulary items would be spoken, and by extension the various hypotheses could be given a priori probabilities proportional to the probabilities of occurrence of their associated vocabulary strings, normalized if necessary to account for the many-one correspondence between hypotheses and utterances. This procedure is, however, unsound, as it can only be justified by the improbable assumption that different hypotheses within a given graph are associated with vocabulary strings which are equally likely to give rise to a directed graph isomorphic to the given one; i.e., that graph structure and utterance content are statistically independent!

in the earlier LCSR environment, where utterances of any length and all digits could be expected, it was natural to treat all hypotheses as equally likely a priori. An alternative to this assumption, which turned out to have remarkably little impact on recognition accuracy, was to assume that the a priori probability of a given hypothesis H in a graph G being correct is the product of the probabilities that H correctly lables each potential recognition either "real" or "artifact" correctly (recognitions in H are labeled "real", all other "artifact"), and that the probability that any particular potential recognition is real or artifactual depends only on its machine type, and is revealed by its rate of occurrence in real or artifactual form in the training sample. This leads to

$$Prob(H) = \prod_{\pi \in H} A_r(m(\pi)) \prod_{\pi \notin H} A_a(m(\pi))$$

where A_a and A_r are probability values on the set M of machine types. This method of estimating the a priori probability of H has no rigorous basis and may differ considerably from the truth. It seems, in particular, to make use of the same data as are used in computing the probability of occurrence of association between various types of machines, and may be a redundant introduction of the same type of effects.

A rigorous approach to estimating the a priori probabilities of various hypotheses might start with accumulating examples represented <u>by a given graph</u>, and observing the relative frequency with which individual hypotheses are correct in this collection. But this computation is not practical, as the number of graphs far exceeds the number of input utterances, which is itself a large number. Even if the computation could be performed, how to represent,store and retrieve the resulting data is a difficult problem. In the absence of such an approach, a simple theory for the rate of occurrence of hypotheses and graphs could be used. I have not been able to posit such a theory in a satisfying way; i.e., in a way which leads to a simple and plausible calculation. I believe any thoroughly satisfactory solution will entail revision of the concepts of association and the potential predecessor relation, as well as the nature of an hypothesis.

In the AIC environment, however, some unquestionable conclusions can be reached about the relative a priori probabilities of various hypotheses. In fact, it is precisely in this factor that the information unique to the AIC environment is most naturally used. The details are discussed later, in connection with the new informatior

<u>How MINT Solves the Problem</u>

As described in the LISTEN final report, MINT solves the problem of finding the hypothesis in the graph G with highest posterior probability by a dynamic programming procedure. Crucial to the derivation of that procedure is the fact that the problem can be converted into the task of finding the hypothesis/path in G with minimum "cost", where the cost is a sum of values associated with each potential recognition and edge in an hypothesis. These individual costs are negative logarithms of ratios of probabilities described above. As the MINT procedure for AIC will be a modification of the current version, we must review the definition of these costs and how they are computed. The description below differs somewhat from that given in the final report, but the same object is being described in both cases. (The description given here may make the connection between LISTEN's data base and voice data collection and processing a little clearer.)

The following definitions are useful.

As described above in connection with inter-word delays and overlays, the probability of occurrence of a given value of delay is assumed to depend upon whether or not it is a delay between real recognitions, and also on the types of machines responsible for the recognitions it joins. The same concept is extended to initial and final delays. If we let the expression " $e_{ij} \in H$ " indicate T or F depending upon whether the edge e_{ij} joining nodes n_i and n_j is part of the hypothesis H, we are assuming the existence of a function, p_d , of four variables such that

$$\mathrm{Prob}(\mathcal{D}(e_{ij}) = d|H) = p_d(d, m(n_i), m(n_j), e_{ij} \in H).$$

(The arguments of p_d are integers, two extended machine types and either T or F.)

The probability of occurrence of a given quality category for a potential recognition is assumed to depend only upon whether or not that potential recognition is real or artifactual. Using the expression " $\pi \in H$ " to denote T or F according as π is, or is not, in H, it is natural to define the function p_q of two variables such that

$$\mathrm{Prob}(V(\pi)=q|H) = p_q(q, \pi \in H).$$

The probability that a potential recognition π has a particular T counter history qualtiy value $t(\text{in } \mathcal{J})$ is assumed to be a function of the machine type responsible for π and whether or not π is real. We therefore define the function p_t of three variables such that

$$\mathrm{Prob}(Q_T(\pi)=t|H) = p_t(t, m(\pi), \pi \in H).$$

Treating L-counter history values in similar fashion we define p_ℓ by

$$\mathrm{Prob}(Q_L(\pi)=\ell|H) = p_\ell(\ell, m(\pi), \pi \in H).$$

The probability that a potential recognition π, caused by machine type m, has an associated potential recognition (one or more) caused by machine type m', is assumed to depend upon the machine types m and m', and whether or not π is real. In the notation given previously, it becomes natural to define the function p_a of three variables such that

$$\mathrm{Prob}(\mathcal{A}(\pi)=A|H)=\prod_{m\in A}A_h(m)\prod_{m\notin A}(1-A_h(m))$$

where $\quad A_h(m)=p_a(m, m(\pi), \pi\in H)$.

The problem of picking the most probable hypothesis can now be transformed to the problem of finding the least cost path through G as follows. Using the probabilistic model described earlier, we have

$$\begin{aligned}-\ln \mathrm{Prob}(H|\Omega) = -\ln \mathrm{Prob}(H) &+ \ln \mathrm{Prob}(\Omega)\\ &-\sum_{e_{ij}\in E}\ln \mathrm{Prob}(\mathcal{D}(e_{ij})=d_{ij}|H)\\ &-\sum_{\pi_i\in\Pi}(\ln I(q_i)+\ln J(t_i)+\ln K(\ell_i))\\ &-\sum_{\pi_i\in\Pi}\Big(\sum_{m\in A_i}\ln A_h(m)+\sum_{m\notin A_i}\ln(1-A_h(m))\Big)\end{aligned}$$

where $\quad e_{ij}$ is the edge joining n_i to n_j

d_{ij} is the delay noted for edge e_{ij}, and

q_i, t_i, ℓ_i are the intrinsic characteristics observed for potential recognition π_i

$A_i=\mathcal{A}(\pi_i)$, the set of machine types associated with π_i

Using the functions defined above,

$$\begin{aligned}
-\ln \mathrm{Prob}(\Omega|H) = {} & -\ln \mathrm{Prob}(H) + \ln \mathrm{Prob}(\Omega) \\
& -\sum_{e_{ij}\in E} \ln p_d(d_{ij}, m(n_i), m(n_j), e_{ij} \in H) \\
& -\sum_{\pi_i \in \Pi} \{ \ln p_q(q_i, \pi_i \in H) + \ln p_t(t_i, m(\pi_i), \pi_i \in H) \\
& \qquad + \ln p_\ell(\ell_i, m(\pi_i), \pi_i \in H) \\
& \qquad + \Big[\sum_{m\in A_i} \ln p_q(m, m(\pi_i), \pi_i \in H) + \\
& \qquad \sum_{m \notin A_i} \ln (1 - p_q(m, m(\pi_i), \pi_i \in H)) \Big] \} .
\end{aligned}$$

Perhaps the most important thing about this expression is that it entails sums over <u>all</u> edges and potential recognition nodes in the graph, and can be shown to differ by a constant from a sum just over the edges and nodes of the hypothesis H. The constant is

$$\begin{aligned}
K = {} & \sum_{e_{ij}\in E} \ln p_d(d_{ij}, m(n_i), m(n_j), F) \\
& + \sum_{\pi_i \in \Pi} \{ \ln p_q(q_i, F) + \ln p_t(t_i, m(\pi_i), F) + \ln p_\ell(\ell_i, m(\pi_i), F) \\
& \qquad + \Big[\sum_{m\in A_i} \ln p_a(m, m(\pi_i), F) + \sum_{m \notin A_i} \ln (1 - p_a(m, m(\pi_i), F)) \Big] \} \\
& + \ln \mathrm{Prob}(\Omega) .
\end{aligned}$$

whence

$$-\ln \mathrm{Prob}(H|\Omega) = -\ln \mathrm{Prob}(H)$$

$$-\sum_{e_{ij}\in H}\left\{\ln \frac{p_d(d_{ij}, m(n_i), m(n_j), T)}{p_d(d_{ij}, m(n_i), m(n_j), F)}\right\}$$

$$-\sum_{\pi_i\in H}\left\{\ln \frac{p_q(q_i, T)}{p_q(q_i, F)} + \ln \frac{p_t(t_i, m(\pi_i), T)}{p_t(t_i, m(\pi_i), F)} + \ln \frac{p_\ell(\ell_i, m(\pi_i), T)}{p_\ell(\ell_i, m(\pi_i), F)}\right.$$

$$\left. + \left[\sum_{m\in A_i} \ln \frac{p_a(m, m(\pi_i), T)}{p_a(m, m(\pi_i), F)} + \sum_{m\notin A_i} \ln \frac{(1-p_a(m, m(\pi_i), T))}{(1-p_a(m, m(\pi_i), F))}\right]\right\}$$

$$+K.$$

Defining the cost to be associated with each edge and node in the obvious way,

$$-\ln \mathrm{Prob}(H|\Omega) = \sum_{e_{ij}\in H} c_{ij} + \sum_{\pi_i\in H} c_i - \left[K + \ln \mathrm{Prob}(H)\right].$$

The hypothesis with maximum posterior probability of being correct is thus the path of least total cost when all the hypotheses have equal a priori probability of occurring, as in that case the term in brackets in the expression above is identical for all hypotheses.

When the alternative treatment of a priori probability described earlier is imposed, one proceeds as follows. Under the alternative treatment, the a priori probability of a hypothesis H is assumed to be

$$\mathrm{Prob}(H) = \prod_{\pi\in H} B_r(m(\pi)) \prod_{\pi\notin H} B_a(m(\pi))$$

as described in the discussion of the difficulties connected with a priori probabilities. Here B_r and B_a are two probability distributions on the set of machine types. Defining

$$c_i' = -\ln \frac{B_r(m(\pi_i))}{B_a(m(\pi_i))}$$

and

$$K' = K + \sum_{\pi_i\in\Pi} \ln B_a(m(\pi_i)),$$

one obtains $$-\ln \mathrm{Prob}(H|\Omega) = \sum_{e_{ij}\in H} c_{ij} + \sum_{\pi_i\in H} (c_i + c_i') - K'.$$

So the hypothesis with maximum posterior probability is again the one with minimum total cost, after adjusting the cost associated with each potential recognition in a very simple way.

To illustrate the mechanics of the procedure, and the significance of the statistical models of the variables, consider the computation of the cost associated with an edge e_{ij} . The models given before indicate that the probability of occurrence of a particular delay value depends upon distribution parameters p_o and λ_o if e_{ij} is incident at Start, or W and μ if e_{ij} is incident at two potential recognitions. Each of these distribution parameters is assumed to be dependent upon the machine type responsible for the potential recognitions at which the edge is incident, and whether or not the edge is real. When e_{ij} is incident at End, the distribution of delay values depends upon parameters we shall denote W_e and μ_e if the edge is real, and W_e' and μ_e' if the edge is not real. The parameters W_e through μ_e' are all dependent upon the machine type of the potential recognition at which e_{ij} is incident. Using the symmetric truncated exponential distribution forms described earlier, it can be seen that

$$c_{ij} = \begin{cases} \left.\begin{array}{l} \alpha_j \text{ if } d_{ij} = 0 \\ \beta_j + \gamma_j d_{ij} \text{ if } d_{ij} \geq 2 \end{array}\right\} \text{if } n_i = \text{Start} \\ \phi_i + \text{Max}\left(0, \frac{|d_{ij} - \mu_e|}{w_e} - 1\right) \text{ if } n_j = \text{End} \\ \chi_{ij} + \text{Max}\left(0, \frac{|d_{ij} - \mu_r|}{w_r} - 1\right) - \text{Max}\left(0, \frac{|d_{ij} - \mu_a|}{w_a} - 1\right) \text{ ow.} \end{cases}$$

where

$$\alpha_j = -\ln \frac{p_o(m(n_j), T)}{p_o(m(n_j), F)}$$

$$\beta_j = -\ln \frac{\lambda(m(n_j), T)(1 - p_o(m(n_j), T))}{\lambda(m(n_j), F)(1 - p_o(m(n_j), F))}$$

$$\gamma_j = \lambda(m(n_j),T) - \lambda(m(n_j),F)$$

$$\phi_i = \ell n\left(\frac{2w_e}{m_e w_e'}\right)$$

$$w_e = w_e(m(n_i))$$

$$w_e' = w_e'(m(n_i))$$

$$\mu_e = \mu_e(m(n_i))$$

m_e is the normalization factor associated with w_e and μ_e

$$x_{ij} = \ell n\left(\frac{w_r}{w_a}\right)$$

$$w_r = w(m(n_i), m(n_j), T)$$

$$w_a = w(m(n_i), m(n_j), F)$$

$$\mu_r = \mu(m(n_i), m(n_j), T)$$

$$\mu_a = \mu(m(n_i), m(n_j), F).$$

These equations reflect the assumption that the distribution of end delay values is uniform, with density $1/w_e'$, over the entire interval of interest.

The matrix GAP in MINT's data base contain numbers analogous to α, β, γ, ϕ, μ_e, w_e, λ, μ_r, w_r, μ_a and w_a, from which the cost C_{ij} is computed.

The cost associated with each node is computed in similarly simple fashion. There are contributions due to the nodes' observed quality category, T-counter statistic Q_T, L-counter statistic Q_L, and set of associated machine types, A. A term is computed for each of these contributions, as a function of the observed data and sometimes of the machine type responsible for the node.

Since the probability of occurrence of quality category is assumed to be independent of machine type, the term contributed by quality category is a function of quality category only. A function ψ of quality category is therefore used, where for quality category q,

$$C_q(\pi) = -\ln \frac{P_q(q(\pi),T)}{P_q(q(\pi),F)} = \psi(q(\pi)).$$

The log likelihood ratio for T-counter statistic is assumed to be a quadratic function of the T-counter statistic value, up to a certain limit beyond which it is assumed to be a constant. The parameters are machine-type dependent. The contribution due to T-counter values is therefore computed as

$$C_t(\pi) = -\ln \frac{p_t(t(\pi),m(\pi),T)}{p_t(t(\pi),m(\pi),F)} \cong \begin{cases} \tau_0(m(\pi)) + \tau_1(m(\pi))t + \tau_2(m(\pi))t^2 & \text{if } t = t(\pi) \leq \lambda_T(m(\pi)) \\ \tau_3(m(\pi)) & \text{if } t > \lambda_T(m(\pi)) \end{cases}$$

The machine-type dependent parameters τ_0 thru λ are estimated from Q_T values observed in training data by a rather complicated process.

The log likelihood ratio for the L-counter statistic Q_L can be computed from the assumed distribution of Q_L, which is a mass concentration at zero, and an exponential distribution over positive values of Q_T. The distribution is dependent upon machine type and whether or not the potential recognition is real; this leads to the computation

$$c_\ell(\pi) = -\ln \frac{p_\ell(\ell(\pi), m(\pi), T)}{p_\ell(\ell(\pi), m(\pi), F)} = \begin{cases} \rho_0(m(\pi)) \text{ if } \ell = 0 \\ \rho_1(m(\pi)) + \rho_2(m(\pi))\ell \text{ if } \ell > 0 \end{cases}$$

Finally, the log likelihood ratio of associations among recognitions of various machine types is stored as a matrix of values

$$\theta_1(m,m') = -\ln \frac{p_a(m,m',T)}{p_a(m,m',F)} + \ln \frac{(1-p_a(m,m',T))}{(1-p_a(m,m',F))}$$

and a vector of values

$$\theta_2(m') = -\sum_{m \in M} \ln \frac{(1-p_a(m,m',T))}{(1-p_a(m,m',F))},$$

which facilitate the computation of cost due to association, as it can be found by a number of additions equal to the number of associated machine types, as

$$c_a(\pi) = \theta_2(m(\pi)) + \sum_{m \in A(\pi)} \theta_1(m, m(\pi)).$$

MINT finds the hypothesis of minimum cost by keeping a pointer for each node indicating the immediate predecessor in the lowest cost path through that node. The additive property of the cost allows one to find this "optimal predecessor" by considering only the total cost of the optimal paths leading to each potential predecessor, and the cost of the edge in common with the potential predecessor. To implement the process it is thus necessary to keep both the pointer to the optimal predecessor, and the total cost of the optimal path from there to Start. Let C_i^* be the cost of the optimal path from Start to the node n_i . Then the recursive property of costs which is exploited in this dynamic programming algorithm is

$$C_i^* = C_i + \min_{\substack{n_j \text{ is a potential} \\ \text{predecessor of } n_i}} [C_{ji} + C_j^*]$$

and of course the optimal predecessor of n_i is the node n_j which minimizes the cost ($C_{ji} + C_j^*$). When End is finally reached, the optimal path through the whole graph can be recovered by following the optimal predecessor links upward.

New Information in the AIC Environment

The utterances to be processed by LISTEN in the AIC nevironment differ from those encountered in the LCSR project in that they consist of a combination of non-numerical and numerical data. The special features of this environment allow one to state a priori, with high probability, that these utterances consist of an initial segment of non-numerical data and a final segment consisting of 3 digits, representing a number between 000 and 359, inclusive.

In addition, the Isolated Word Recognition capability will be used to estimate, with something like 90% accuracy, the last digit spoken.

MEX will be operating throughout the utterance, and no doubt many artifactual digit recognitions will occur during the non-digital portion of the utterance.

The kinds of knowledge available for choosing the best explanation of an utterance in the AIC environment depend upon the voice data collection procedures to be followed, as well as the vocal behavior imposed by the AIC task. In particular, it is assumed that voice data will be collected and processed for AIC trainees much as it was for LCSR subjects. The training utterances (for connected word recognition) will consist of connected strings of digits, and not, for instance, the non-digital portions of the AIC vocabulary. This precludes measuring the frequency with which various digits are falsely recognized in the non-digital portion of these utterances - a datum which could, in principle, be used in the recognition procedures.

In terms of the directed graph of an utterance, the problem of picking the best explanation for the utterance can again be interpreted as choosing the path through the graph which maximizes the product of a priori probability of occurrence and conditional probability of producing the observed data. The a priori data, derived from the nature of the AIC environment, include the facts that these are probably at least three digits in the utterance, constituting a number between 000 and 359, that any additional recognitions are probably false, the three real digits are the last in the utterance, and that the last digit is probably the one specified by the IWR subsystem.

As only digits will be spoken during the training period, it will be assumed that nothing is known about artifactual recognitions which may occur during the non-digit portions of an utterance. Although some data of this sort could be collected during operation, this adaptive type of approach will not be adopted now. We shall instead base the selection of the best explanation of an utterance exclusively on the characteristics of the last three potential recognitions in each hypothesis, ignoring entirely the characteristics of the remainder of the hypothesis, except for its duration, as explained below.

The start time of a potential recognition is particularly important when that potential recognition is being considered as a candidate for the first digit spoken, as in that case the start time is equal to the length of time it took to speak the initial non-digital portion of the utterance. In the AIC environment there are several possible non-digital initial utterance segments, such as "Come left to heading...". These utterances will be recognized and discriminated by the IWR portion of SUS. Furthermore, the mean time to speak the non-digital portions of these utterances will also be known to the IWR subsystem, and can be used as an additional factor in discriminating among hypotheses, by comparing it to the start time of a hypothetical first spoken digit.

The basic dynamic programming scheme used in MINT to find the best explanation of an utterance does not apply in the AIC environment (at least not without modification) because individual nodes do not have unique optimal predecessors. To illustrate this fact, consider a potential recognition whose corresponding vocabulary item is "7". This potential recognition, like others, may lie in several hypotheses. Suppose it has potential predecessors corresponding to vocabulary items "2" and "3", and that the "3" is intrinsically a more attractive potential recognition, on the basis of nominal violations and very nominal T and L Counter values, for instance. For a hypothesis in which the "7" is the next-to-last digit spoken, the "3" is an unlikely immediate predecessor because the associated interpretation is a number greater than 359. However, for a hypothesis in which the "7" is the last digit spoken, the "3" is an acceptable immediate predecessor if it has an immediate predecessor of suitable type (0,1,2, or 3). The optimal predecessor of a node is thus dependent upon the position of the node in an hypothesis. Treating it as such solves the problem, as will be shown.

Formal Basis for New MINT

To describe new AIC version of the MINT algorithm, we extend the formal definition of the problem to include the directed graph and the elements of the observation (the functions $\mathcal{D}$, $\mathcal{L}$ and $\mathcal{A}$) described before, and add a function which gives the start time of each potential recognition (measured in "counts" from the beginning of the utterance). This is a function from the set Π of potential recognitions into the set of natural numbers:

$$\mathcal{S}: \Pi \to \mathbb{N}$$

The observation, Ω , is now a 4-tuple($\mathcal{D}, \mathcal{L}, \mathcal{A}, \mathcal{S}$), rather than the 3-tuple ($\mathcal{D}, \mathcal{L}, \mathcal{A}$).

As we expect only three digits to be spoken, and those at the end of the utterance, it is appropriate to modify the definition of an hypothesis to consist of a path in the directed graph descriptive of the utterance, not from Start to End as before, but from a potential recognition, through two others to End. As there is some finite probability that a trainee may incorrectly speak other than three digits, it might seem appropriate to also consider hypotheses with other than three digits, but with lower a priori probability. There is definite merit to this viewpoint, but in the interest of simplicity we shall restrict attention to three-digit hypotheses. This should seldom lead to erroneous speech <u>understanding</u> in the AIC environment, although it will definitely lead to erroneous speech <u>recognition</u> when other than three digits are spoken. The reason for this is that the output of the speech recognition procedures (a three digit string) will be compared with what was supposed to have been spoken, and it is very unlikely that the new MINT will produce the "correct" response when it was not in fact given by the trainee. Perhaps the most likely case leading to such erroneous "correct" response would arise when the trainee speaks four or more digits consisting of the correct three digits, preceeded by some extraneous digits. By considering four (or more) digit hypotheses the new MINT would have some potential to detect this kind of erroneous response. However, the data structure and the processing time grow with the maximum number of digits to be considered, and the added potential is not worth the added implementation cost.

Implementation costs are also the basis for restricting attention to responses on the interval 000-359. In reality, magnetic headings are given as numbers on the interval 001-360; that is, magnetic north is indicated by the string 360 rather than 000. MINT is intrinsically concerned with contiguous pairs of potential recognitions, as it uses the potential predecessor relation, and the cost associated with an edge (and an edge is really a potential contiguous pair of recognitions), so it is natural and easy to deal with the interval 000-359, as it can be associated with the set of all hypotheses whose first contiguous pair correspond to numbers in the interval 00-35. Special logic and perhaps special data structures have to be introduced to treat correctly the real interval, 001-360, and as only one value of the 360 possible is concerned, it is not worth the added implementation burden to accommodate the real interval.

Hypotheses will thus consist of three potential recognitions and the edges connecting them. The changes of definition of the observation and a hypothesis in no way invalidates the statistical decision theory approach or its solution through the application of Baye's theorem. We still select the hypothesis with maximum posterior probability;

$$P(H|\Omega) = \frac{P(H)P(\Omega|H)}{P(\Omega)},$$

and as the probability of the observation is common to all hypotheses, we are again led to detect the hypothesis with maximum product of a priori probability and probability of producing the observation.

The a priori probability of each hypothesis will be modeled as the product of three probabilities; one depending upon whether the hypothesis corresponds to a number in the interval 000-359, one depending upon whether the last potential recognition in the hypothesis agrees with the IWR prediction of the last vocabulary item, and one depending upon the a priori probability used in the older LISTEN system. (As discussed earlier, there are two versions of a priori probability in the current system.) Thus the new a priori probability of a hypothesis containing potential recognitions π_i, π_j, π_k in that order is

$$P(H) = P_I(m(\pi_i), m(\pi_j), m(\pi_k)) P_{IWR}(m(\pi_k)) P^*(H)$$

where $$P_I(m_1, m_2, m_3) = \begin{cases} p_I & \text{if } m_1 m_2 m_3 \text{ corresponds to a number in 000-359} \\ 1-p_I & \text{otherwise.} \end{cases}$$

$P_{IWR}(m)$ is the apriori probability that the last digit spoken was of type m. Received from the IWR subsystem).

$P^*(H)$ is the currently computed a priori probability of H.

Notice that p_I is simply the a priori probability that the trainee will in fact respond with a number in the interval 000-359. Also notice that in this formulation a vector of probabilities, $P_{IWR}(m)$, is assumed to be provided by the IWR subsystem, indicating its estimated probability that the last digit spoken is a particular vocabulary item. This may be zero (or at least vanishingly small) for all but one machine type, or any other distribution over the set of machine types.

As before, we model the conditional probability of the observation $\Omega = (\mathcal{D}, \mathcal{L}, \mathcal{A}, \mathcal{S})$ as the product of the condition probabilities of its components, thus invoking the assumption of independence among these components:

$$\mathrm{Prob}(\Omega|H) = \mathrm{Prob}(\mathcal{D}|H)\,\mathrm{Prob}(\mathcal{L}|H)\,\mathrm{Prob}(\mathcal{A}|H)\,\mathrm{Prob}(\mathcal{S}|H)$$

The first three factors, dealing with the delays between nodes and the properties of each potential recognition, will be modeled exactly as before. The last term, dealing with the start time of potential recognitions will be modeled as follows. The mean start time of recognition of the first spoken digit will be assumed to be the same as the mean time required to speak the non-digital portion of the utterance, and will be supplied by the IWR subsystem. Let this value be μ_{ST}. Then the distribution of start times for recognition of the real first digit spoken will be assumed to be (discretized) normal with mean μ_{ST} and standard deviation equal to a fraction f (about 20%) of μ_{ST}. The distribution of start times for all recognitions (real or artifactual) which are not the real first digit spoken will be assumed to have the constant density $1/\mu_{ST}$ throughout the region of interest. Finally, the probability of observing all of the start times for all the potential recognitions will be assumed to be the product of the probabilities of observing each individually. That is

$$\mathrm{Prob}(\mathcal{S}|H) = \prod_{\pi \in \Pi} p(\mathcal{S}(\pi)|H)$$

where

$$p(\mathcal{S}(\pi)=x|H)=\begin{cases}\frac{1}{f\mu_{ST}\sqrt{2\pi}}\,e^{-\left(\frac{x-\mu_{ST}}{f\mu_{ST}}\right)^2} & \text{if } \pi \text{ is the first recognition in } H\\ \frac{1}{\mu_{ST}} & \end{cases}$$

Proceeding exactly as before, by introducing negative natural logarithms and defining costs C_{ij} associated with edge e_{ij} and C_i associated with potential recognition π_i (each cost being the negative natural logarithm of a likelihood ratio), one obtains for hypothesis H consisting of recognitions π_i, π_j, π_k,

$$-\ln \mathrm{Prob}(H|\Omega) = -\ln P_I(m(\pi_i),m(\pi_j)) - \ln P_{IWR}(m(\pi_k)) + C_i''$$

$$+\sum_{e_{ij}\in H} c_{ij} + \sum_{\pi_k\in H} c_k - \left[K + \ln P^*(H)\right]$$

where all terms except C_i'' are as previously defined. The term C_i'' is the log likelihood ratio corresponding to the start time observed for the first recognition in H, π_i ;

$$c_i'' = -\ln\left[\frac{1}{f\mu_{ST}\sqrt{2\pi}}\,e^{-\left(\frac{\mathcal{S}(\pi_i)-\mu_{ST}}{f\mu_{ST}}\right)^2}\Big/(1/\mu_{ST})\right]$$

$$= A + (B\mathcal{S}(\pi_i) - C)^2$$

where $A = \ln f\sqrt{2\pi}$

$B = 1/(f\mu_{ST})$

and $C = 1/f$

The term in brackets in the expression for $\ln(H|\Omega)$ is again either constant for all hypotheses, or by suitable modification of the cost associated with <u>every</u> node, it can be made to be so. Hence, it can be neglected in our search for the best hypothesis.

The solution is still the hypothesis which minimizes the cost, but the cost now has three new terms. It is natural to interpret these three new terms as costs associated with parts of the hypothesis. We will use the letter a to denote these costs since they arise in the AIC environment. The first term,

$$a_{12} = -\ln P_I(m(\pi_i), m(\pi_j))$$

is naturally interpreted as a cost associated with the edge joining the first and second potential recognitions. Its value is

$$a_{12}(i,j) = \begin{cases} -\ln p_I & \text{if } m(\pi_i), m(\pi_j) \text{ correspond to} \\ & \text{a number in } 00\text{-}35 \\ -\ln(1-p_I) & \text{otherwise.} \end{cases}$$

The second term,

$$a_3(k) = -\ln P_{IWR}(m(\pi_k))$$

is naturally interpreted as a cost associated with the third potential recognition of an hypothesis. Its value is the negative natural logarithm of the IWR-supplied probability that the last digit spoken was of machine type $m(\pi_k)$. The third term,

$$a_1(i) = c_i''$$

is more or less naturally interpreted as a cost associated with the first potential recognition of an hypothesis.

The problem reduces then to finding the hypothesis with minimum cost C, where

$$C(\pi_i, \pi_j, \pi_k) = a_1(i) + a_{12}(i,j) + a_3(k) + \sum_{e_{ij} \in H} c_{ij} + \sum_{\pi_k \in H} c_k .$$

The critical and difficult change in the problem is that the cost associated with an edge or a potential recognition now depends upon its position within an hypothesis. The crucial property of costs which underlay the dynamic programming solution used in the first MINT (described in detail on pages 71 and 72 of the LISTEN final report) no longer holds true. The necessary modification is to associate not one optimal cost and pointer, but three optimal costs and two pointers, with each node. The single optimal cost used before was the total cost of the minimal cost path from Start to the node in question; that is, the cost of the best partial solution leading to that node. The three optimal costs in the new formulation are the total costs of the best partial solution leading to the node in question if it is regarded as the first, the second or the third digit spoken. These costs are different, and in general the optimal predecessors will be different nodes when a node is considered as the second or the third node of an hypothesis. Thus two pointers are needed.

The algorithm required is then quite obvious. The best hypothesis can be identified as the potential predecessor of End which has the smallest optimal cost of the third kind after adding the cost associated with the final delay (the edge incident at End and the node in question). It is found by dynamic programming as before, using the three costs. If we denote the three optimal costs associated with each node as q_1^*, q_2^* and q_3^*, one proceeds by processing each potential recognition, π_i , as it is received from MEX as follows:

1. The optimal path through this node as the first word of an hypothesis is just the node itself. Therefore the total cost of the optimal path of length 1 through this node is just the intrinsic cost of the node (as computed before, reflecting violations, T and L Counters, associations and possibly an a priori consideration) plus its unique cost due to its being the first digit of the hypothesis; thus

$$q_1^*(\pi_i) = a_1(i) + c_i$$

2. An optimal path of length two ending at this node consists of this node, its predecessor and the edge joining them. The total cost of such a partial solution is the sum of the normal costs associated with the edge and this node, together with the optimal cost of the predecessor as the first word-of an hypothesis, and the special cost associated with an edge joining the first two words of an hypothesis (to bias in favor of the interval 00-35). Thus

$$q_2^*(\pi_i) = c_i + \underset{\substack{\pi_j \text{ potential} \\ \text{predecessor of } \pi_i}}{\mathrm{Min}} \left[q_1^*(\pi_j) + a_{12}(j,i) + c_{ji}\right]$$

If, however, this node has only Start, and no potential recognition, as potential predecessor, this node is not a viable candidate as the second member of an hypothesis. Its cost, q_2^* , in this case should be very large. This effect can be obtained by setting q_1^*(Start) to a very large number. The minimization above can then be carried out over all potential predecessor nodes, regardless of whether they are potential recognitions or Start.

The pointer p_2 (which points to optimal predecessor) is set to point at a node which minimizes q_2 .

3. Similarly, an optimal path of length three ending at node π_i is a path of length two, plus an edge plus this node. Its cost is the sum of the normal cost plus the cost reflecting the IWR bias towards certain last words. The result is

$$q_3^*(\pi_i) = c_i + a_3(i) + \underset{\substack{n_j \text{ a potential} \\ \text{predecessor of } \pi_i}}{\text{Min}} \left[q_2^*(n_j) + c_{ji}\right].$$

Again the possibility that the node has no potential predecessor viable as the second word of an hypothesis is nicely handled by setting q_2^*(Start) to a high value. The pointer, p_1 , is determined in the search for minimum q_3 , just as p_2 was.

As stated above, the optimal hypothesis is identifiable as the potential predecessor of End whose q_3^* value, when added to the cost of the edge joining it to End, is minimal. Searching for this optimal hypothesis can therefore be accomplished in much the same way as optimal costs and pointers are determined for any other node. It is more natural, however, to modify MINT to treat the processing of End as a special case.

APPENDIX B

CURVE FITTING ROUTINE

The curve fitting routine SQUISH is designed to fit a curve through the observed points of the cumulative distribution of the Q_T quality function values for both real and artifact recognitions.

More specifically, the observed points which must be fit are points of the cumulative distribution

$$P_R(\delta) = \text{Prob } [Q_T(\pi) < \delta \mid \pi \text{ is real}]$$

for a real recognition, and

$$P_A(\delta) = \text{Prob } [Q_T(\pi) < \delta \mid \pi \text{ is an artifact}]$$

for an artifact.

To do this "fitting", three separate problems must be handled:

(1) An appropriate functional form for the "fitting" function must be found.

(2) The standard or norm of approximation (e.g. least squares, minimax) must be chosen.

(3) An algorithm to accomplish the fitting must be designed.

After several false starts and dead ends, a solution to (1) was found. The fitting function chosen was the three parameter function

$$Q(\delta) = 1 - e^{-\alpha e^{\beta\delta - \gamma\delta^{-2}}}$$

where α, β, and γ are positive real numbers to be determined.

The solution of (2) was to use a minimax approximation. That is, the fit must have the property that the maximum deviation between each pair of observed value Y_i and predicted value $Q(\delta_i)$ is minimal. In other words, the fit is to be an aproximation in the supremum norm.

The algorithm employed to do the fitting — that is, to find the coefficients α, β, and γ for each set of data — relies upon the observation that the sort of minimax approximation which is sought must have a special property. Namely, there must exist exactly four points δ_1, δ_2, δ_3, and δ_4 at which the maximum errors occur, and the maximum errors must have the further property that they are equal in absolute value and alternating in sign. So if Y_i is the observed value at δ_i and $Q(\delta_i)$ is the predicted value at δ_i, then the errors $\varepsilon_i = Y_i - Q(\delta_i)$ must satisfy

$$\varepsilon_1 = -\varepsilon_2 = \varepsilon_3 = -\varepsilon_4$$

The actual fitting procedure begins by finding initial guesses at the α, β, and γ parameters, determining at each stage the four points of worst fit, and then refining the choices of α, β, and γ until minimax is achieved.

The typical error at minimax so far observed is on the order of ± Ø.Ø4.

Another curve fitting routine, QDFIT, is designed to fit a parabola to computed values of the negative log-likelihood ratio

$$-\ln\left(\frac{Q'_{\text{real}}(\alpha, \beta, \gamma; \delta)}{Q'_{\text{artifact}}(\alpha_\emptyset, \beta_\emptyset, \gamma_\emptyset; \delta)}\right)$$

The values α, β, γ, $\alpha_\emptyset$, $\beta_\emptyset$, and $\gamma_\emptyset$ are provided by SQUISH for each set of real and artifact data. The procedure used in QDFIT is a simple least squares fit of a parabola to the computed log-likelihood ratios.

DISTRIBUTION LIST

Naval Training Equipment Center
Orlando, FL 32813 34

Defense Documentation Center
Cameron Station
Alexandria, VA 22314 12

Headquarters
Air Training Command, XPT
Attn: Mr. Don Meyer
Randolph AFB, TX 78148 2

All other addressees receive one copy.

US Air Force Human Resources Lab
AFHRL-SMS
Computational Sciences Div.
Statistical and Computer
Technology Branch
Brooks AFB, TX 78235

Dr. Raj Reddy
Professor, Dept of Computer Science
Carnegie-Mellon University
Pittsburg, PA 15213

Dr. E. Cohen
Link Division
The Singer Co.
Binghamton, NY 13902

Mr. Harry Whitted
Electronics Engineer
Naval Oceans Systems Center
271 Catalina Blvd
San Diego, CA 92152

Mr. Leon A. Ferber
Vice President
Perception Technology Group
95 Cross St
Winchester, MA 08190

Mr. Thomas B. Martin
President, Threshold Technology, Inc.
1829 Underwood Blvd
Delran, NJ 08075

Dr. Clayton R. Coler
Research Scientist
NASA Ames Research Center
Mail Stop 239-2
Moffett Field, CA 94035

Mr. Melvin L. Strieb
Program Manager
Human Factors Analytics
2500 Maryland Road
Willow Grove, PA 19090

Dr. Edward A. Stark
Link Division
The Singer Co.
Binghamton, NY 13902

Scientific Advisor
HQ US Marine Corps
Washington, DC 20380

Mr. Donald W. Connolly
ANA-238
Research Psychologist
Federal Aviation Administration
FAA/NAFEC
Atlantic City, NJ 08504

Chief of Education and Training
Liaison Office
Human Resource Laboratory
Flying Training Div
Attn: CAPT W. C. Mercer
Williams AFB, AZ 85224

Director Education Development
Academic Computing Center
US Naval Academy
Annapolis, MD 71402

Mr. Charles W. Geer
Engineer
The Boeing Co.
P. O. Box 2999, MS 82-87
ORG 2-3541
Seattle, WA 98124

Mr. Richard W. Obermayer
Navy Personnel Research and Development Center, Code 34
San Diego, CA 92152

Commanding Officer
Naval Aerospace Medical Research
Laboratory
Naval Air Station
Pensacola, FL 32508

National Defense College
of Canadian Army
Staff College
Ft. Frontena
Kingston, Ontario, Canada

AFHRL/FTO
ATTN: Mr. R. E. Coward
Luke AFB, AZ 85309

Prof. J. Allen, Room 36-575
Massachusetts Institute of
Technology
Cambridge, MA 02139

Headquarters
Air Training Command, XPT
Attn: Dr. Don Meyer
Randolph AFB, TX 78148

Commandant
USA Field Artillery School
Target Acquisition Dept
Attn: Eugene C. Rogers
Ft. Sill, OK 73503

Dr. Donald E. Walker
SRI International
Menlo Park, CA 94025

Mr. Warren Lewis
Human Engineering Branch
Naval Ocean Systems Ctr.
Code 8231
San Diego, CA 92152

Mr. Larry L. Pfeifer
Vice President
Signal Technology, Inc.
15 W. De La Guerra
Santa Barbara, CA 93101

Dr. Jesse Orlansky
Institute for Defense Analyses
Science & Technology Div
400 Army-Navy Drive
Arlington, VA 22202

Seville Research Corp.
Suite 400, Plaza Bldg.
Pace Blvd. at Fairfield
Pensacola, FL 32505

Mr. Bruce T. Lowerre
Computer Scientist
Systems Control, Inc.
1801 Page Mill Road
Palo Alto, CA 94304

Lt. Col. Robert L. Hilgendorf, USAF
Aeronautical Systems Div/AERS
Wright Patterson AFB, OH 45433

Mr. Horace Enea
President, Heuristics, Inc.
900 N. San Antonio Road
Los Altos, CA 94022

US Air Force Human Resources
Lab/DOJZ
Brooks AFB, TX 78235

Head, Research Development and
Studies Branch (OP-102)
Office of Deputy Chief of Naval
Operations
(Manpower, Personnel and Training,
OP-01)
Washington, DC 20350

Mr. Don Murray
Telcom Systems, Inc.
320 West Street Road
Warminister, PA 18974

Chief of Naval Education and
Training Support
Pensacola, FL 32509

MAJ Neal D. Morgan (LGY)
USAF Logistics Management Center
Gunter AFS, AL 36114

Dr. Beatrice Oshika
System Development Corp.
2500 Colorado Avenue
Santa Monica, CA 90406

Mr. N. Rex Dixon
Speech Processing Consultant
IBM, Thomas J. Watson Research Ctr
P.O. Box 218
Yorktown Heights, NY 10098

Library, Navy Personnel Research
and Development Center
San Diego, CA 92152

Commanding Officer
Rome Air Development Center
Document Library (TSLD)
Griffiss AFB, NY 13440

Chief
ARI Field Unit
P.O. Box 476
Fort Rucker, AL 36362

Mr. Arthur W. Lindberg
Electronics Engr.
US Army Avionics
R&D Activity, DAVAA-E
Ft. Monmouth, NJ 07703

Mr. Robert S. Hartman
VP Electronics, Gould Inc.
Hydrosystems Div.
125 Pinelawn Rd
Melville, NY 11746

USAHEL/AVSCOM
Dir., RD&D
ATTN: DRXHE-AV (Dr. Hofmann)
P.O. Box 209
St. Louis, MO 63166

NASA Scientific & Technical
Information Facility
P.O. Box 8757
ATTN: Acquisitions
BWI Airport, MD 21240

Scientific Technical Information
Officer, NASA
Washington, DC 20546

US Air Force Human Resouces Lab
AFHRL-AS
Advance Systems Division
Wright-Patterson AFB, OH 45433

Dr. Klaus Lindenberg
Dir. of Advanced Systems
Appli-Mation, Inc.
3191 Maguire Blvd. #244
Orlando, FL 32803

Commanding Officer
Naval Air Technical Training Ctr.
NAS, Memphis 85
Millington, TN 38054

Mr. I. James Whitton, Systems Engineer
General Electric - AES
831 Broad Street MD 700
Utica, NY 13503

Mr. J. Michael Nye, President
Marketing Consultants International, Inc.
100 W. Washington St., Suite 216
Hagerstown, MD 21740

Mr. Marvin B. Herscher, Executive
Vice President
Threshold Technology, Inc.
1829 Underwood Blvd.
Delran, NJ 08075

Mr. Kieffer Hart, President
Telcom Systems, Inc.
2300 S. 9th Street
Arlington, VA 22204

Mr. Roland Payne, Program Manager
Systems Control, Inc.
1801 Page Mill Road
Palo Alto, CA 94304

US Air Force Human Resources Lab
AFHRL-TT
Technical Training Division
Lowry AFB, CO 80230

Calspan Corp.
Librarian
P. O. Box 400
Buffalo, NY 14225

Mr. John C. Simons
Human Factors Engineering Services Gp
Systems Research Labs
2800 Indian Ripple Road
Dayton, OH 45440

Mr. Don F. McKechnie
Research Psychologist
Aerospace Medical Research Lab
Human Engineering Division
Wright-Patterson AFB
Dayton, OH 45433

Dr. Mark F. Medress
Manager, Speech Communications
Sperry Univac Defense Systems
Speech Communications Dept
Univac Park, P.O. Box 2525
UOP16
St. Paul, MN 55165

Director, Human Engineering Lab
USA Aberdeen Research Dev. Ctr
Attn: Mr. C. A. Fry, DRXHE-HE
Aberdeen Proving Grounds, MD
21005

Dr. Danny Cohen, US
USC Information Sciences Institute
4676 Admiralty Way
Marine Del Rey, CA 90291

Mr. Emmett L. Herron
Human Factors Engineer
Bunker Ramo Corporation
4130 Linden Ave., Suite 302
Dayton, OH 45432

Commander
Navy Air Force, US Pacific Fleet
NAS North Island (Code 316)
San Diego, CA 92135

Commander
Naval Air Force, US Pacific Fleet
NAS North Island
San Diego, CA 92135

Capt. Barry P. McFarland
USAF, ASD/ENECH
Wright Patterson AFB
Dayton, OH 45433

Mr. Sam S. Viglione
Interstate Electronics
Ave.
Anaheim, CA 92803

Mr. Bob Davis
Speech Systems Research
Systems and Information Sciences Lab
Texas Instruments
P. O. Box 5936
Dallas, TX 75222

AFHRL/FTO
Attn: Mr. R. E. Coward
Luke AFB, AZ 85309

Chief of Naval Operations
OP-987H
Attn: Dr. R. G. Smith
Washington, DC 20350

Mr. Tom Balcer
Manufacturing Engineer
Lockheed Missles and Space Co.
P. O. Box 504, B151 0/8616
Sunnyvale, CA 94088

Commander
Naval Air Systems Command
(AIR 340F)
Washington, DC 20361

Chief of Naval Operations
Attn: CAPT H. J. Connery, OP-506H1
Washington, DC 20350

Commander
Training Command
Attn: Educational Advisory
US Pacific Fleet
San Diego, CA 92147

US Army Research Institute for the
Behavioral and Social Sciences
Attn: Brian Kottas
Fort Knox Field Unit
Fort Knox, KY 40121

Chief of Naval Operations
OP-07
Washington, DC 20350

Mr. R. S. Dunn
NASA, Ames Research Center
Mail Stop 207-5 HQUSATRL
Moffett Field, CA 94035

Mr. Robert Osborn
VP Engineering
Dialog Systems, Inc.
32 Locust Street
Belmont, MA 02178

Mr. Lon Sorenson
Systems Engineer
SEMCOR, Inc.
Strawbridge Lake Office Blvd
Route 38
Moorestown, NJ 08057

Director
Southern Field Division
Office of Civilian Personnel
Building A-67
Attn: Jim Herndon
NAS
Norfolk, VA 23511

Mr. Jared J. Wolf
Senior Scientist
Bolt, Beranek & Newman, Inc.
50 Moulton Street
Cambridge, MA 02138

AFHRL/PE
Brooks AFB, TX 78235

Dr. Wayne A. Lea
Research Linguist
Speech Communications Research Lab
806 W. Adams Blvd.
Los Angeles, CA 90007